UNIVERSITY OF CALIFORNIA
RIVERSIDE

Deep Models for Image Analysis, Synthesis and Scene Perception

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Runze Li

December 2023

Dissertation Committee:

Dr. Bir Bhanu, Chairperson
Dr. Chinya V. Ravishankar
Dr. Ahmed Eldawy
Dr. Tamar Shinar

The Dissertation of Runze Li is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I thank my dissertation chair, Dr. Bir Bhanu, who never gives up on motivating me to finish my degree. I would like to thank Dr. Chinya V. Ravishankar, Dr. Ahmed Eldawy and Dr. Tamar Shina for being my committee members on my dissertation proposal and defense. I would also want to thank Dr. Subir Ghosh for conferring my doctoral candidacy. I am grateful for all colleagues and friends who were part of my graduate life. I would like to extend my heartfelt thanks to my special friend, Zion, the lovely cat who has been my companion throughout the last two years of my graduate life, sharing both joys and tears with me.

To my mother Li Huyan, my father Jinke Li, my grandfather Jintang Hu, my grandmother Xiaofeng Zhang and my wife Dr. Shuang Wei for all the love and support.

ABSTRACT OF THE DISSERTATION

Deep Models for Image Analysis, Synthesis and Scene Perception

by

Runze Li

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2023
Dr. Bir Bhanu, Chairperson

Visual analysis is a fundamental task to develop approaches to understand contents. With the advances of deep learning models, visual image synthesis plays an increasingly important role because it leverages generative models for synthesizing novel images which can be used for training and testing. This dissertation presents new techniques for visual analysis and synthesis by developing novel deep learning techniques, in particular methods for sports analytics, vision and language, face synthesis, scene perceptions, human body mesh understanding and visual interpretations of generative models. This dissertation deals with sports analytics, image and language pre-training, translation of facial attributes, passive depth estimation in the indoor environments, human body mesh reconstruction and visual interpretations of variational autoencoders. It develops a range of advanced techniques for dribbling and goal recognition, language-supervised contrastive learning for visual understanding, translating features on a human face, passive range application of AR/VR, reconstructing human body mesh and interpreting variational autoencoders. Both theory and experimentation will be presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Rich and complex events in sports have led to the development of a wide-variety of techniques for interpreting content of sports videos in terms of players' actions, poses, gait, performance, etc. This is due to the requirements from coaches, trainers and players who expect to analyze actions in top sports events, as well as sports fans who practice to imitate professional playing skills, *e.g.*, dribbling, shooting, etc. However, this poses two key challenges for automated sports analysis community. Firstly, there are extremely limited public sports datasets. Secondly, recent advances in interpretations of sports activities, *e.g.*, soccer, are predominantly made through analyzing coarse-grained contents. Players' fine-grained skills analysis still remains under-explored. To alleviate these problems, this thesis (a) collects the dataset of highlight videos of soccer players, including two coarse-grained action types of soccer players and six fine-grained actions of players. Detailed annotations are provided for the collected dataset, in terms of action classes, bounding boxes, segmentation maps, and body keypoints of soccer players, and positions of a soccer ball in a game. (b) leverages the understanding of complex highlight videos by proposing an energy-motion fea-

tures aggregation network-***EMA-Net*** to fully exploit energy-based representation of soccer players movements in video sequences and explicit motion dynamics of soccer players in videos for soccer players' fine-grained action analysis. Experimental results and ablation studies validate the proposed approach in recognizing soccer players actions using the collected soccer highlight video datasets.

Large-scale vision and language pre-training has made great progress in recent years. This thesis presents RECLIP (Resource-efficient CLIP), a simple method that minimizes computational resource footprint for CLIP (Contrastive Language Image Pretraining). Inspired by the notion of coarse-to-fine in computer vision, we leverage small images to learn from large-scale language supervision efficiently, and finetune the model with high-resolution data in the end. Since the complexity of the vision transformer heavily depends on input image size, our approach significantly reduces the training resource requirements both in theory and in practice. Using the same batch size and training epoch, RECLIP achieves highly competitive zero-shot classification and image-text retrieval accuracy with 6 to 8$\times$ less computational resources and 7 to 9$\times$ fewer FLOPs than the baseline. Compared to the state-of-the-art contrastive learning methods, RECLIP demonstrates 5 to 59$\times$ training resource savings while maintaining highly competitive zero-shot classification and retrieval performance. Finally, RECLIP matches the state of the art in transfer learning to open-vocabulary detection tasks, achieving 32 APr on LVIS. We hope this work will pave the path for the broader research community to explore language supervised pretraining in resource-friendly settings.

Synthesis of face images by translating facial attributes is an important problem in computer vision and biometrics and has a wide range of applications in forensics, entertainment, etc.

Recent advances in deep generative networks have made progress in synthesizing face images with certain target facial attributes. However, visualizing and interpreting generative adversarial networks (GANs) is a relatively unexplored area and generative models are still being employed as black-box tools. This thesis takes the first step to visually interpret conditional GANs for facial attribute translation by using a gradient-based attention mechanism. Next, a key innovation is to include new learning objectives for knowledge distillation using attention in generative adversarial training, which result in improved synthesized face results, reduced visual confusions and boosted training for GANs in a positive way. Firstly, visual attentions are calculated to provide interpretations for GANs. Secondly, gradient-based visual attentions are used as knowledge to be distilled in a teacher-student paradigm for face synthesis with focus on facial attributes translation tasks in order to improve the performance of the model. Finally, it is shown how "pseudo"-attentions knowledge distillation can be employed during the training of face synthesis networks when teacher and student networks are trained to generate different facial attributes. The approach is validated on facial attribute translation and human expression synthesis with both qualitative and quantitative results being presented.

Self-supervised monocular depth estimation has seen significant progress in recent years, especially in outdoor environments, *i.e.*, autonomous driving scenes. However, depth prediction results are not satisfying in indoor scenes where most of the existing data are captured with hand-held devices. As compared to outdoor environments, estimating depth of monocular videos for indoor environments, using self-supervised methods, results in two additional challenges: (i) the depth range of indoor video sequences varies a lot across different frames, making it difficult for the depth network to induce consistent depth cues for training, whereas the maximum distance in

outdoor scenes mostly stays the same as the camera usually sees the sky; (ii) the indoor sequences recorded with handheld devices often contain much more rotational motions, which cause difficulties for the pose network to predict accurate relative camera poses, while the motions of outdoor sequences are pre-dominantly translational, especially for street-scene driving datasets such as KITTI. In this thesis, we propose a novel framework-***MonoIndoor++*** by giving special considerations to those challenges and consolidating a set of good practices for improving the performance of self-supervised monocular depth estimation for indoor environments. First, a depth factorization module with transformer-based scale regression network is proposed to estimate a global depth scale factor explicitly, and the predicted scale factor can indicate the maximum depth values. Second, rather than using a single-stage pose estimation strategy as in previous methods, we propose to utilize a residual pose estimation module to estimate relative camera poses across consecutive frames iteratively. Third, to incorporate extensive coordinates guidance for our residual pose estimation module, we propose to perform coordinate convolutional encoding directly over the inputs to pose networks. The proposed method is validated on a variety of benchmark indoor datasets, *i.e.*, EuRoC MAV, NYUv2, ScanNet and 7-Scenes, demonstrating the state-of-the-art performance. In addition, the effectiveness of each module is shown through a carefully conducted ablation study and the good generalization and universality of our trained model is also demonstrated, specifically on ScanNet and 7-Scenes datasets.

Recent works have made significant advances of skeleton-based 3D human mesh reconstructions. In this thesis, we consider the problem of estimating frame-level full human body meshes given a video of a person with natural motion dynamics. While much progress in this field has been in single image-based mesh estimation, there has been a recent uptick in efforts to infer mesh dy-

namics from video given its role in alleviating issues such as depth ambiguity and occlusions. However, a key limitation of existing work is the assumption that all the observed motion dynamics can be modeled using one dynamical/recurrent model. While this may work well in cases with relatively simplistic dynamics, inference with in-the-wild videos presents many challenges. In particular, it is typically the case that different body parts of a person undergo different dynamics in the video, e.g., legs may move in a way that may be dynamically different from hands (e.g., a person dancing). To address these issues, we present a new method for video mesh recovery that divides the human mesh into several local parts following the standard skeletal model. We then model the dynamics of each local part with separate recurrent models, with each model conditioned appropriately based on the known kinematic structure of the human body. This results in a structure-informed local recurrent learning architecture that can be trained in an end-to-end fashion with available annotations. We conduct a variety of experiments on standard video mesh recovery benchmark datasets such as Human3.6M, MPI-INF-3DHP, and 3DPW, demonstrating the efficacy of our design of modeling local dynamics as well as establishing state-of-the-art results based on standard evaluation metrics.

Recent advances in convolutional neural network (CNN) interpretability have led to a wide-variety of gradient-based visual attention techniques for generating visual attention maps. However, most of these methods require a classification-type design architecture, and consequently concenrate on classification/categorization-type tasks. Extending these methods to generate visual attention maps for other kinds of computer vision models, e.g., variational autoencoders (VAE) is not trivial. In this paper, we present a method that helps bridge this crucial gap, proposing to compute *VAE attention* as a means for interpreting the latent space learned by a VAE. We first present methods to generate visual attention maps from the learned latent space, and then show how they

can be used in a variety of applications: localizing anomalies in images, including medical imagery, and improved latent space disentanglement. We conduct extensive experiments on a wide-variety of benchmark datasets to demonstrate the efficacy of the proposed VAE attention.

Chapter 2

Energy-Motion Features Aggregation Network for Players' Fine-grained Action Analysis in Soccer Videos

2.1 Introduction

Computer vision techniques have been widely utilized in sports (soccer, ice hockey, basketball, baseball, etc.) for broadcasting, tactical analysis for players' actions and classification and skill/talent recognition and improvement. For instance, Wu *et al.* [442] collected a basketball dataset, NCAA+ to study event classification in basketball games. Xu *et al.* [448] proposed a deep architecture to automatically score the fancy figure skating in videos. Qi *et al.* [333] designed a hierarchical recurrent-neural-network-based framework for sports video captioning and conducted

experiments on volleyball datasets. Kong *et al.* [219] focused on analyzing group dynamics and describing team tactics in volleyball games.

As one of the most popular sports, soccer-related activities have drawn significant interest worldwide. From the perspective of professional activities, Word Cup and European Cup, as world-class tournaments, are held in turn every two years and national soccer teams compete for participation in the final. In Europe, five top football leagues, Premier League, La Liga, Championnat de France de football Ligue 1, Bundesliga and Lega Serie A, organize the highest-level soccer games every year and attract soccer players from around the world to participate. From a commercial perspective [228], the Premier League is the most profitable league, which has achieved a revenue of 6,032 million euros in 2020-2021. The booming business value in soccer has driven a deeper analysis targeting on players-related actions, to obtain precise and elaborate statistics on soccer players. Besides, a number of companies are expanding their business for sports analysis, including Intel, Second Spectrum, Sports Logic, etc. From the perspective of high school participation, soccer has been growing as one of the most popular sports played by high school students. In USA, according to the survey conducted by Statista.com [228], 846,844 high school students have participated in soccer during the year 2019-2020, including 459,077 boys and 394,105 girls. In addition to sports events related research, the analysis of injuries in sports games [304, 324] has become popular and has gained increasing attention because (1) monitoring the health conditions of players can benefit players' performance and (2) predicting potential risks can aid in preventing crises from adversely impacting players in sports. Thus, soccer is an important research area for video technologies to develop analytical tools which extract useful information from a large numbers of soccer videos.

Developing computer vision techniques for analyzing soccer videos has led to a number of research works in the field of circuits and systems for video technology. Baysal *et al.* [20] proposed a method for tracking multiple players in soccer videos by designing a new particle-filter-based model. They also studied coarse-grained jersey number classification in soccer videos. Wang *et al.* [432] focused on action localization in soccer games and proposed leveraging the rich text information to assist in analyzing soccer videos. Theagarajan *et al.* [396] designed a system for detecting ball possession, team identification and tactical statistics generation for players in soccer games. Liu *et al.* [260] proposed universal jersey number detection methods based on deep learning for a wide range of sports. However, from these recent works, it can be observed that: (i) Limited work has been done on video-based soccer players action analysis; (ii) The video data on players are rarely provided and even more rare are the detailed annotations that are provided for players action analysis. This poses challenges for researchers because deep models always require large data with ground-truth information for training; (iii) Most of the existing video analysis work concentrates on players' coarse-grained analysis, *i.e.*, high-level event detection, group activity identification, etc., while players' fine-grained action analysis, *e.g.*, dribbling, shooting, which are important in soccer games, related research work is rare.

In the field of computer vision and circuits and systems for video technology, this is the first paper that provides soccer players' (including goalkeepers) highlight video datasets with multiple fine-grained action classes and develops algorithms for players' fine-grained video analysis.

There are two main problems in performing automated soccer video analysis: The first problem is that most of the existing methods conduct coarse-grained soccer analysis, *e.g.*, event

detection, game recognition, while soccer player's fine-grained analysis remains relatively unexplored. In this work, based on the collected soccer players highlight videos, this paper develops a system for analyzing detailed skills performed by soccer players, in particular, players' fine-grained actions analysis. We propose a novel energy-motion features aggregation network (***EMA-Net***) for soccer players' fine-grained action analysis. The input soccer highlight videos are processed by deep-learning-trained models for detecting and segmenting soccer players, soccer balls and parsing soccer players' body keypoints. Next, the results so obtained are processed further by two core components for players' fine-grained action classification. It is observed that most of the soccer highlight videos are captured without a chance to know camera parameters. Soccer players always move at a fast speed to perform certain skills within seconds and cameras are trying to catch up with players, causing camera motions. A sequence of frames with both temporal and spatial information would be distorted and can provide little information without image registration. To solve above problems, we propose to use a transformation-based approach to register a sequence of frames with target soccer players into a single energy-based image representation. This representation can serve as a way to encode history of players' motions and maintain fine-grained information. In addition, to explicitly encode players' movements in a video sequence, we propose to use a self-attentive motion modelling module to capture and model motion dynamics. Further, we design an energy-motion features aggregation module for players' fine-grained action classification. Unlike many existing transformer works for sports analysis [354, 415] which use the same query, key and value for learning features, we give special considerations on using the registered energy image as query and key features to model players' fine-grained action information and use motion features as value features to explicitly account for the motion dynamics. This allows the model to focus on learning

fine-grained features. We validate the proposed method on extensive dataset and design experiments that show the superiority of performance over the standard action recognition algorithms for soccer players' fine-grained action classification.

The second problem with automated video analysis is the lack of labeled soccer highlight video data. Soccer highlight videos are captured in close-view to record a player's action where the camera is moving to track the player who is controlling the ball and performing various tasks that may require significant skills. These videos illustrate expert skills in soccer games and they are shared by fans around the world. However, there are very few datasets available for research usage, especially for the streaming data which are used by a business. There exist data-in-the-wild that can be obtained from online platforms, for example, Youtube, which are captured without ground-truth annotations. Annotating video data is time-consuming and expensive. To bridge the gap of the shortage of soccer highlight video data with players' fine-grained actions, in this paper, we collect soccer players highlight video datasets. It consists of two coarse-grained actions which are dribbling and shooting, and six fine-grained types of actions, for dribbling they are "Stepover", "Elastico" and "Chop" and for shooting they are "Penalty-Kick shooting", "Goal shooting" and "Free-Kick shooting".

We have provided detailed annotations of players-related contents by using CVAT [367] to annotate each frame in a soccer video and export annotations in various widely-used formats, *e.g.*, COCO object detection, PASCAL VOC segmentation, etc. Besides, we have processed the collected dataset by a number of computer vision algorithms to generate auxiliary annotations, including an object detection tool, Mask R-CNN [161], and one of the most robust human pose

estimation toolboxes, OpenPose [48], which outputs players segmentation maps and keypoints on players' body. It is noted that the goalkeeper is also annotated and processed.

The collected dataset is suitable for performing analysis for different fine-grained actions of soccer players. For instance, investigating dribbling skills in soccer games, which has been beneficial to both the clubs to train their players, and for the defenders to know how to improve their defending skills. For example, top players, like Cristiano Ronaldo, Lionel Messi, Neymar Jr., have been well-known icons for their smart dribbling skill, which has helped them evade through defenders and score in soccer games. Soccer fans have been constantly amazed by adept dribbling skills and have been curious to understand and analyze "What dribbling styles have been shown by the players" when they have watched top-class soccer games like World Cup, UEFA European Champions, etc. The collected dataset and annotations will be released at `https://github.com/bragilee/Soccer_Players_Highlight_Videos_Dataset` under approved licenses after the publication of this paper.

2.2 Related Work and Our Contributions

2.2.1 Deep Learning Models

Deep convolutional neural networks [221, 162] have been widely used for image classification. Girshick *et al.* [138] proposed the Faster R-CNN for object detection. This work has been further optimized as the Mask R-CNN [161] which established a new benchmark in predicting object bounding boxes, segmenting semantic masks. Cao *et al.* [48] concentrated on using deep models in predicting human poses and proposed the OpenPose which achieved high performance both in accuracy and speed. In recent years, action recognition has made significant progress

driven by various learning paradigms with deep models [51, 403, 203, 123, 124, 423, 325]. Besides, research [413, 110, 274] on vision transformers has been very active and they have become new backbone networks with competitive performance. Various ongoing efforts [12, 26, 302, 190] have already studied the applications of transformers in video classification. Another line of research concentrates on skeleton-based action analysis. Yan *et al.* [451] proposed a Spatial Temporal Graph Convolutional Networks for skeleton-based action recognition. Shi *et al.* [372] designed a two-stream adaptive graph convolutional network to fully model the skeleton information from the lengths and directions of bones. Plizzaria *et al.* [325, 326] explored the integration of transformer modules in designing a Spatial-Temporal Transformer network for modelling joints dependencies for action recognition. Instead of simply using skeleton representations, Duan *et al.* [112] proposed to use 3D heatmap volume and investigated the combination with RGB frames for robust skeleton-based action recognition.

2.2.2 Soccer Video Analysis

Computer vision in sports has been a hot topic in recent years and there has been an increasing amount of research emanating from numerous sports events [122, 323, 346, 44, 322, 335, 296, 320, 320, 414, 289, 371, 393, 60, 189]. There have been many ongoing efforts which focus on soccer video analysis including video summarization, event classification and action spotting. Our work is focused on players' fine-grained analysis. It is directed towards in-depth exploration of soccer highlight videos, specifically, with insights on how to recognize dribbling and shooting styles of a player who is controlling the ball and performing smart actions. Most existing player action analysis research belonged to either video-based or image-based methods. Sarkar *et al.* [358] presented an automatic technique to analyze different stages in a valid pass between soccer players

for calculating ball possession statistics from the video of a soccer match. Arbues-Sanguesa *et al.* [10] used offensive player's orientation along with location information and opponents' spatial configuration to generate the feasibility of pass events in soccer games. Theagarajan *et al.* [397, 396] conducted soccer game analysis by identifying a player who has the ball at any moment in soccer matches, recognizing teams and generating tactical performance statistics from soccer videos. Li *et al.* [242] conducted fine-grained action classification for soccer dribbling styles. They manually cropped dribbling player of interests and used the player's keypoints and dribble energy image to classify dribbling styles.

2.2.3 Sports Dataset

Datasets are critical for researchers to conduct sports analysis using deep learning techniques because model training is typically data-hungry. Researchers have developed soccer datasets at varying scales. Soccer Player [283] is a small scale datasest which consists of 2,019 annotated frames with 22,586 player bounding boxes for player detection and tracking. Tsunoda *et al.* [407] have collected a private video dataset captured using 14 synchronized and calibrated Full HD cameras with annotations of bounding boxes for players. This dataset is designed for "Futsal", a football-based game played on a smaller hard court, *e.g.*, indoors. SportsMOT [252] is a dataset for multiple object tracking in sports videos(including basketball, volleyball and soccer) and its soccer category has 680 frames in 21 tracks. SoccerNet [136] dataset includes 550 complete broadcast soccer games and 12 single-camera games taken from major European leagues. This dataset is further extended to SoccerNet-v2 [93], SoccerNet-v3 [79] for action spotting, camera segmentation and boundary detection, and SoccerNet-Tracking [82] for multiple object tracking. As existing datasets

have not fully explored players' fine-grained actions, we have collected soccer highlight video datasets and provided detailed annotations for research purposes.

2.2.4 Contributions of this Chapter

- **First automated system** which integrates player and ball detection, parsing of player key-points and energy-motion features aggregation core modules for soccer players' fine-grained action analysis in soccer videos.

- **A novel energy-motion features aggregation network** (***EMA-Net***) for soccer players' fine-grained analysis, including (a) a novel joints-guided image registration module to transfer a sequence of frames to an image representation which can handle raw video clips at multi-scale resolutions and solve camera motion problems, (b) a self-attentive motion modelling module to explicitly model dynamics of soccer players in a video sequence, and (c) an energy-motion features aggregation module which takes energy features and explicit motion features for players' fine-grained action analysis. This is the first work that fully leverages the design ideas of vision transformer in modelling registered energy and motion features for the analysis of soccer players in videos.

- **Soccer Players Highlight Video Datasets** are collected from various sources for soccer players action analysis. It includes two coarse-grained of widely-performed actions which are dribbling and shooting, and six fine-grained styles of actions, three of which are for dribbling: "Stepover", "Elastico" and "Chop" and the other three are for shooting: "Penalty-Kick shooting", "Goal shooting" and "Free-Kick shooting". Detailed annotations are provided manually or by running third-party algorithms with the collected soccer players highlight

video datasets, including bounding boxes, semantic segmentations, human body keypoints of soccer players and goalkeepers (which has rarely been studied) who are performing certain actions, and bounding boxes of soccer balls.

- **Extensive Experiments** with discussions, ablation studies are conducted for soccer players action analysis and the effectiveness of the proposed approach is validated using soccer players highlight videos. This paper is the first one to conduct experiments with extensive considerations of both the goalkeepers and soccer ball explicitly.

2.3 Technical Approach

In this section, we introduce the overall pipeline called Energy-Motion Features Aggregation Network (***EMA-Net***), which is an end-to-end framework to process incoming soccer players highlight videos, and explain the role of each of the individual components for the analysis of a soccer player. The overall pipeline is shown in Fig. 2.1. Our framework accepts video sequences as the input. The input data are *first* processed by the player and ball detection module which outputs the bounding boxes, segmented maps of soccer players and soccer balls in a video. Soccer players are also parsed with human pose estimation module which outputs kinematic human keypoints of players. *Second*, the soccer players segmentation maps and body keypoints are used for performing the registration of soccer players in a video sequence. It creates an energy image representation of the soccer players in a video sequence that passes through a feature extractor to obtain energy features which will be aggregated in the next stage. *Third*, the sequence of soccer players segmentation maps are taken as input to a self-attentive motion modelling module to encode motion dynamics of soccer players and output motion features. Finally, we design an energy-motion feature aggregation

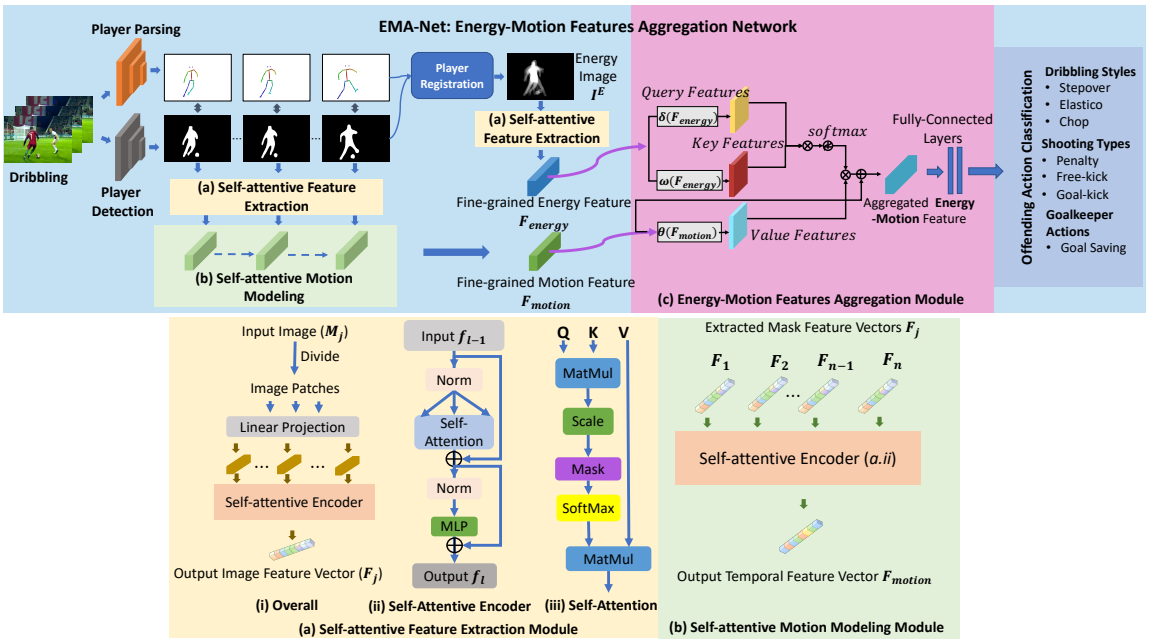

Figure 2.1: The proposed framework. Top: Energy-Motion Features Aggregation Network (***EMA-Net***) for a player's action analysis in soccer highlight video. Bottom: (a) Self-attentive Feature Extraction Module; (b) Self-attentive Motion Modelling Module; (c) Energy-Motion Feature Aggregation Module.

module to aggregate energy features and motion features learned for soccer players' fine-grained action classification.

2.3.1 Localization, Segmentation and Keypoints Parsing

Soccer Player and Soccer Ball Detection and Segmentation: In soccer games, soccer players perform various actions such as dribbling and shooting that require the interaction with soccer ball. Most existing work [396, 397] focuses on analyzing soccer players actions without considering contextual information related to soccer ball. Some ongoing efforts analyze ball possession, passing ball, etc, while knowing soccer ball context in a game is a strong assumption. We

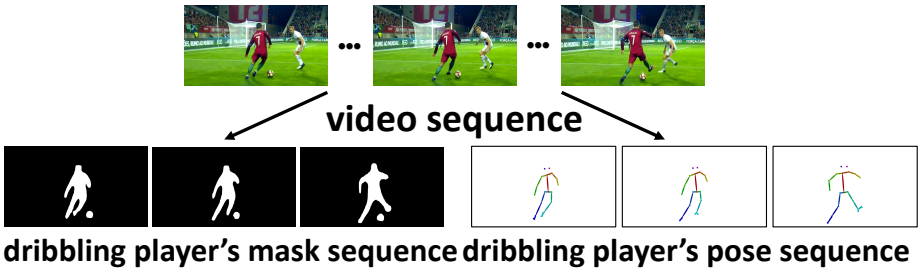

Figure 2.2: Top: video sequence of a dribbling action performed by Cristiano Ronaldo (red jersey). Bottom left: segmentation mask of the dribbling player and soccer ball. Bottom right: pose of the dribbling player.

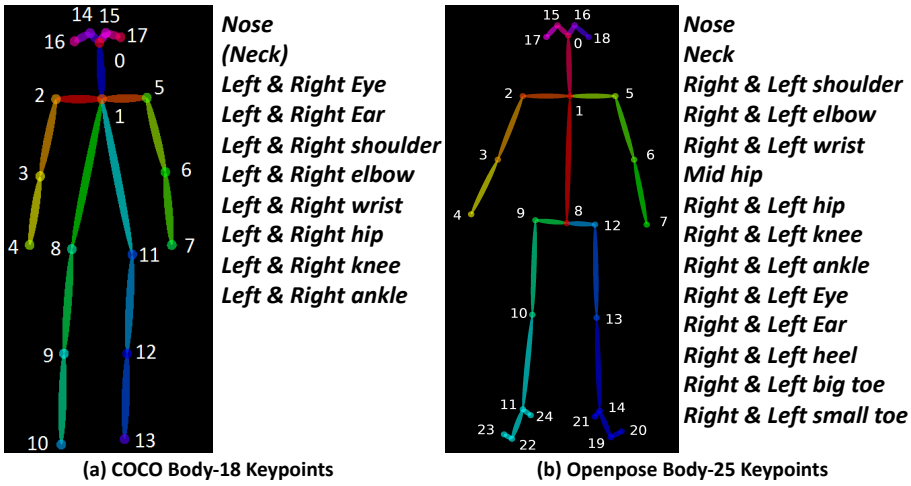

Figure 2.3: COCO 18 keypoints and OpenPose 25 keypoints for human body.

propose to use object detection tools for both soccer players and soccer ball. Existing objection detection tools fall into two main categories, single-stage methods [343, 267] and two-stage methods [139, 138]. Single-stage methods, in particular, YOLO [343] and SSD [267] show advantages in detecting objects at a fast rate while R-CNN [139, 138] and its extension Mask R-CNN [161] have advantage in outputting extensive information, including bounding boxes, segmentation maps and human body keypoints. Because Mask R-CNN is designed with two-stages and outputs several types of predictions, this network runs relatively slower compared with single-stage detectors. Mask

R-CNN [161] also maintains excellent feasibility and it can be easily extended with different output streams for various purposes. In this work, we use the Mask R-CNN [161] to preprocess video sequences to localize soccer players, soccer balls, segment these regions of interests and parse soccer players.

Mask R-CNN designs a network with two-stage procedures. In the first stage, the backbone network proposes candidate object bounding boxes for input images. In the second stage, Mask R-CNN outputs a binary mask for each region-of-interests in parallel to predict the classes with confidence values and bounding boxes. The model that we use was pretrained on the Microsoft COCO dataset [254]. Microsoft COCO dataset consists of diverse images for the class"Person" which also includes sports players and the images in this dataset have different scale variations, and occlusions which are similar to the scenario of a soccer field. Our approach processes every frame of each highlight video through the Mask R-CNN. For a given frame, the bounding boxes belonging to the class "Person" with probability greater than a given threshold are considered to be the locations of the soccer players for that frame. Similarly, the bounding boxes belonging to the class "Sports ball" with probability greater than a given threshold are considered to be the locations of the soccer balls. In our approach, we set the threshold for soccer player to be 0.5 and the threshold for soccer ball to be 0.4, empirically. The bottom left part in Fig. 2.2 shows the segmentation results of the soccer player on a video sequence using Mask R-CNN.

Soccer Player Pose Parsing: We use Keypoint R-CNN [315] which extends Mask R-CNN by adding one stream for predicting human body keypoints for each detected object-"Person". So for each player, by performing detection and pose parsing, we obtain bounding box, segmentation map and body keypoints from the model output given the input video data. For each frame as

the input, the locations of the kinematic keypoints are predicted as one of the outputs. As the model is trained on COCO dataset, there are 18 keypoints in the output which are shown in Fig. 2.3(a).

As it can be seen from the Fig. 2.3(a), keypoints of lower torso are quite sparse and some important joints are not included, in particular, the "mid hip" joint, and this could potentially raise problems in soccer players action analysis due to the missing skeleton connected by the "mid hip" and "neck" joint. It is observed that lower torso plays a significant role for soccer players to perform fancy soccer actions at a fast speed (normally within 1 second). Thus, rich joints information of lower torso can be helpful in observing, analyzing and recognizing soccer actions played by soccer players. To alleviate the sparse output from Keypoint R-CNN predictions, we propose to use OpenPose [48] as an auxiliary tool to extract 2D pose information of soccer players. OpenPose takes a color image of size $w \times h$ as the input and produces the 2D locations of anatomical keypoints for each person in the image as the output. The output from the OpenPose contains 25 keypoints, including rich joints in the lower torso, which is shown Fig. 2.3(b). After obtaining two human keypoints outputs, we take keypoints on hips which support the main torso of human body and compensate each other if they are not detected by one of the models because we would like to ensure we can perform hip-joints based image registration. If keypoints of hips are not obtained, the keypoints of shoulders will be used for image registration. If none of the keypoints on hips or shoulders are outputted, the keypoints of knees will be utilized. Image registration will not be performed if none of keypoints on the torso or keens are detected. To this end, for soccer players of interests in each frame of soccer videos, segmentation map, keypoints of soccer players and soccer ball are obtained, which are shown in Fig. 2.2. We have focused on a player's fine-grained action analysis based on detection, segmentation and keypoints results. We match keypoints and bounding boxes

using the following steps: (1) First, we run Mask R-CNN and Keypoints R-CNN models available in PyTorch [315] on each frame, which provide outputs that consists of bounding boxes and keypoints of detected players. (2) Second, we infer the bounding box according to the coordinates of keypionts that correspond to each detected player. (3) Third, we match the inferred bounding boxes from the keypoints and the detected bounding boxes between consecutive frames. We calculate the mean IOU between the two bounding boxes so obtained and the matching is successful when the mean IOU is larger than a pre-set threshold. After many experiments, for soccer dribbling videos, the threshold was empirically set at 0.8 and for soccer shooting videos, the threshold was empirically set at 0.5. (4) Finally, we use the matched bounding boxes and keypoints of a player in each frame for the subsequent processing.

The keypoints we consider to perform the energy-based player image registration include, left and right shoulders, left, mid and right hips and left and right knees. The intuition for selecting these keypoints for image registration is that it is observed that hip, shoulder and knee areas are mainly supporting a soccer player's body movements when the player is performing fancy skills. Specifically, for dribbling and shooting actions, soccer players are swiftly moving their feet to demonstrate their professional skills. During a player's body movement, we argue that the hip area of a player's body performs core functions in stabilizing the gesture of the whole body. Therefore, we primarily select hip joints to process the proposed image registration to obtain registered energy image which would be introduced in the next section.

2.3.2 Energy-Motion Features Aggregation Network for Soccer Player Action Classification

We design three core modules: (1) an image registration module to register a soccer player in a video sequence into one fine-grained energy image representation, (2) a self-attentive feature extraction and motion modelling module to explicitly encode fine-grained motion dynamics of soccer players, (3) an energy-motion features aggregation module to encode fine-grained energy features and explicit motion features for players' fine-grained action classification in soccer players highlight videos. Details of each module is explained in the following sub-sections.

B.1 Soccer Player Image Registration

Energy-based Player Image Registration: In soccer highlight videos, players demonstrate certain soccer skills in a sequence of frames, which causes both spatial motion of objects-of-interests within each frame and camera motion across consecutive frames and obtaining camera information is expensive for researchers. One line of research work [51] focuses on utilizing a spatial stream and a temporal stream to process a sequence of frames and such methods always face bottlenecks in how to extract valid information with two streams to train a network and they need large-scale data during the training. Another line of research finds efficient ways to encode dynamics of a sequence of frames. Bilen *et al.* [32] introduce a dynamic image that summarizes the appearance and dynamics of an entire video sequence for video analysis. Bobick *et al.* [34] propose binary motion-energy image and motion-history image to represent where motion has occurred and its intensity in an image sequence. Han *et al.* [157] propose the gait energy image as a spatio-temporal gait representation for recognition of walking humans. These energy-based representations can serve

as simple yet efficient ways for representing basic human actions, for example, walking, running, etc. Nevertheless, simply adapting these representations to the professional skills demonstrated by soccer players fails as both cameras and soccer players may be moving in different directions at a fast speed. In this paper, we take an initial step for a player's image registration to obtain an energy image representation for complex player's motions in highlight videos. By using the player's energy image, we expect to encode the fine-grained information of a soccer player's moving sequence into a single image representation. In addition, we consider elimination of influences imposed by challenging camera motions by proposing a registration method that transforms a sequence of frames into the same embedding so as to generate a single energy image representation for the soccer player of interest in a highlight video.

Algorithm 1 Transformation-based Player's Image Registration for Energy Image Representation Generation

Input: Detected player's mask images $M_j, j \in 1, 2, 3, ..., n$ and parsed keypoints results $P_j, j \in 1, 2, 3, ..., n$ of the soccer player of interest in each highlight video.

Output: Registered player's mask images $\overline{M_j}, j \in 1, 2, 3, ..., n$ of the player of interest in each highlight video.

From $M_j, j \in 1, 2, 3, ..., n$, find the index k where the pose results contain valid keypoints. Take the left hip, right hip and the left shoulder for an example, the keypoints are described as (x_k^{lh}, y_k^{lh}), (x_k^{rh}, y_k^{rh}) and (x_k^{ls}, y_k^{ls}) along (x, y) axes, respectively.

for $j = 1$ *to n frames in the sequence and* $j \neq k$ **do**

localize left hip, right hip and left shoulder keypoints of the soccer player, which are described as (x_j^{lh}, y_j^{lh}), (x_j^{rh}, y_j^{rh}) and (x_j^{ls}, y_j^{ls}), respectively calculate the transformation matrix T_A following equation (2.1) by using the triangle Δ_j constructed from (x_j^{lh}, y_j^{lh}), (x_j^{rh}, y_j^{rh}) and (x_k^{ls}, y_k^{ls}) and the triangle Δ_k constructed from (x_k^{lh}, y_k^{lh}), (x_k^{rh}, y_k^{rh}) and (x_j^{ls}, y_j^{ls}) calculate the center point C_j of the line connected by the left hip and right hip keypoints and the aligned center point of the line $C_j^{'}$ with transformation matrix T_A as $C_j^{'} = T_A \times C_j$ based on the $C_j^{'}$ and C_j, register M_j to obtain registered mask image $\overline{M_j}$.

Transformation-based Image Registration: It is based on the player detection and keypoints parsing results from Section 2.3.1. The registration process is described in **Algorithm** 1.

The equation for calculating transformation matrix is given below:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \times \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{2.1}$$

where (x_1, y_1) is the end point from the triangle Δ_k and (x_2, y_2) is the end point from the triangle Δ_j, which is shown at the bottom of Fig. 2.4. Once we obtain a sequence of registered mask images $\overline{M_j}, j \in 1, 2, 3, ..., n$ for the soccer player of interest for each video clip $V_i, i \in 1, 2, 3, ..., N$, we calculate the energy image representation I_i^E. The energy image representation results of transformation-based image registration is illustrated on the right side of Fig. 2.4.

Thanks to the player's image registration module, our system does not perform any camera calibration and the camera operator is allowed to freely pan, tilt and zoom the camera depending on where the action is happening on the soccer field. For instance, the field-of-view of cameras is narrow when capturing dribbling and goal shooting actions performed by soccer players because cameras are moving at a fast speed. Similarly the field-of-view of cameras is wide when the soccer players are performing Free-Kick shooting in soccer field. Our system can handle these situations by means of player detection process and players image registration operations.

In this paper, we concentrate on fine-grained analysis of dribbling and shooting skills performed by soccer players so hip joints are considered with the highest priority to perform the player's image registration and we use keypoints parsing results from two sources, Keypoint R-CNN and Openpose which are shown in Fig. 2.3, to increase chances in obtaining valid hip joints. However, in situations when hip joints are missing, for example, only one left hip joint is valid, we take the joint from neck, left shoulder or left knee to perform registration because joints of neck and

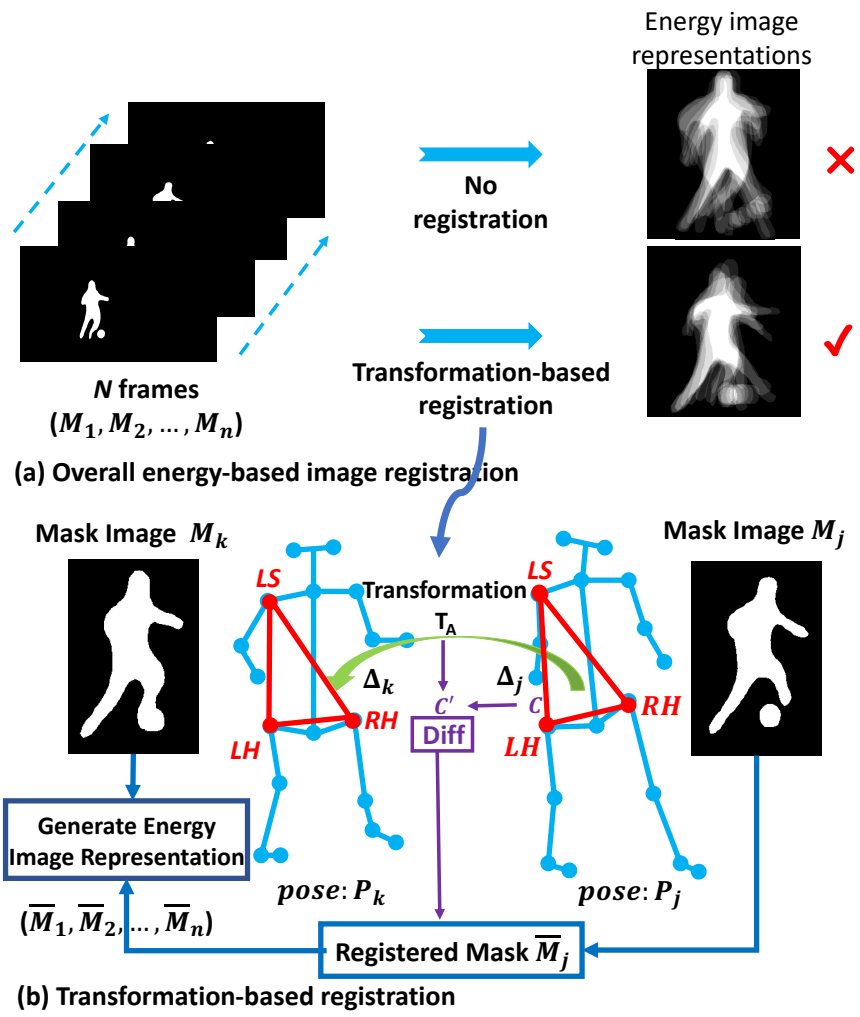

Figure 2.4: Comparisons of the energy image representations with and without image registration. LH, RH, LS: joints of left hip, right hip and left shoulder. Δ_k, Δ_j: triangles with coordinates which are used to calculate the transformation.

shoulders still remain in the main torso of the human body and joints of knee are the closest to the hip area.

B.2 Self-attentive Processing Modules

1) Self-attentive Feature Extraction: Most of work conducting analysis on soccer players in images and videos uses convolutional neural networks to extract features with multi-level kernels designed

for high-level content analysis, *e.g.*, event detection, action spotting, etc. Inspired by recent advances in natural language processing (NLP) in using self-attention layers for modelling language tasks, vision transformers have been used widely for object detection, image classification, etc., resulting in competitive performance [110]. In this paper, we propose to use a self-attention module for extracting energy features from soccer players highlight videos and for modelling soccer players motion dynamics explicitly, outputting fine-grained motion features, which can be aggregated with fine-grained energy features for soccer players action classification.

We use the vision transformer backbone [110] for designing feature extraction module. To elaborate, different from the standard transformers which are designed to accept $1D$ input, the vision transformer divides the image $X \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened $2D$ patches $x_p \in \mathbb{R}^{Q \times (P^2 \times C)}$, where (H, W) is the resolution of the image, C is the number of channels, P^2 is the resolution of each image patch and $Q = HW/P^2$ is the resulting number of patches, which is also the length of effective input sequence for the vision transformer. Firstly, patch embedding layer is designed to project the image patches to a pre-defined-dimensional embedding space. Next, a self-attentive encoder which is composed of a series of self-attention layers with normalization layers is used to learn features from image patches. Finally, multi-layer perceptron (MLP) layers are used to output features learned from the input image. The self-attentive feature extractor is shown in Fig. 2.1(a).

We use this self-attentive feature extraction module to process both registered players energy image representations after image registration module and player segmentation map after detection with Mask R-CNN over the soccer players videos so that parameters can be shared for multi features extraction purposes. For instance, given segmentation mask maps $M_j, j = 1, 2, 3, ..., n$

as the input, the output of self-attentive feature extraction module can be denoted as $F_j, j = 1, 2, 3, ..., n$; given one energy image I_i^E from each soccer highlight video clip $i, i \in 1, 2, 3, ..., N$ as the input, the output can be denoted as F_{energy}, where n is the number of frames and N is the number of videos.

2) Self-attentive Motion Modelling: The self-attentive feature extraction module enables the extraction of features given any types of input images, including energy image, video frames, players segmentation maps and cropped patches of soccer players, etc. In this section, we introduce how we design self-attentive motion modelling module by using the features extracted from each player's segmentation map in a video sequence and it is shown in Fig. 2.1(b). As aforementioned, segmentation map of each frame is denoted as $M_j.j \in 1, 2, 3, ..., n$ in a video sequence $V_i.i \in 1, 2, 3, ..., N$. The output of each frame after feature extraction operation is denoted as $F_j, j = 1, 2, 3, ..., n$, which is designed as the input to the self-attentive motion modelling module. Inspired by the intuition behind the transformer networks [110] where the input image is divided into patches and the patch sequence is composed for further processing. We extend the self-attention module in modelling features sequence $F_j, j = 1, 2, 3, ..., n$ where each item in the sequence represents features extracted from segmentation map $M_j, j \in 1, 2, 3, ..., n$. These operations can be denoted as:

$$f_0 = [F_1; F2; \ldots; F_j], j = 1, 2, 3, ..., n \tag{2.2}$$

$$f_l' = SAE(f_{l-1}) + f_{l-1}, l = 1...L \tag{2.3}$$

$$f_l = MLP(f_l) + f_l', l = 1...L \tag{2.4}$$

$$F_{motion} = Norm(f_l) \tag{2.5}$$

where SAE is self-attentive encoder, f_l is features from the self-attentive layer $l(l = 1, 2, 3, ..., L)$ and $L = 6$ (6 layers are used in our experiments), MLP is multi-layer perception module, $Norm$

is a normalization layer and F_{motion} is the outputted fine-grained motion features of soccer players in each highlight video clip. Following the backbone network in [110], positional encoding is also utilized in Equation (2.2). The self-attentive motion modelling module explicitly encodes the motion dynamics of soccer players who perform certain fine-grained actions across the video sequence. The self-attention module learns motion features with context information that spans long duration. Our expectation is that motion dynamics of players' fine-grained actions can be encoded as fine-grained motion features which can be aggregated with fine-grained energy features and the aggregated features so obtained can contribute collaboratively to the classification of soccer players' fine-grained actions.

B.3 Learning Energy-Motion Feature for Classification

We propose to design a energy-motion feature aggregation module which takes the fine-grained motion features and fine-grained energy features outputs a player's action classification prediction. Motion features are directly obtained from the motion modelling module which encodes motion features of fine-grained actions played by soccer players in soccer highlight videos. Fine-grained energy features are the output of the feature extraction module (see Fig. 2.1) given the registered energy image of a player. We give special considerations to using the registered energy image as query and key features to learn players' fine-grained action information and using motion features as value features to explicitly acccount for the motion dynamics. This can allow the model to focus on learning fine-grained features.

The proposed energy-motion features aggregation module takes the energy features $\mathcal{F}_{energy} \in \mathbb{R}^{N \times D_e}$ and motion features $\mathcal{F}_{motion} \in \mathbb{R}^{N \times D_t}$ as the input to compute aggregated

energy feature, and then the aggregated energy-motion features are used for soccer players' fine-grained actions classification. The aggregated energy-motion features are given by

$$\begin{aligned} \mathcal{F}_{energy-motion} &= \mathcal{F}_{motion} + \\ & g(\delta(\mathcal{F}_{energy}), \omega(\mathcal{F}_{energy}))\theta(\mathcal{F}_{motion}) \end{aligned} \tag{2.6}$$

where δ, ω and θ are the projection functions for the query, key and value, which are given by

$$\begin{aligned} \delta(\mathcal{F}_{energy}) &= \mathbf{W}_{query}\mathcal{F}_{energy}, \\ \omega(\mathcal{F}_{energy}) &= \mathbf{W}_{key}\mathcal{F}_{energy}, \\ \theta(\mathcal{F}_{motion}) &= \mathbf{W}_{value}\mathcal{F}_{motion}, \end{aligned} \tag{2.7}$$

where $\mathbf{W}_{query}$, $\mathbf{W}_{key}$ and $\mathbf{W}_{value}$ are learnable parameters. The g function consists of two operations which are matrix multiplication and softmax, respectively. The aggregated energy-motion features will be flattened and used for soccer players' fine-grained action classification. We apply two fully-connected layers with normalization and DropOut layers and the output is denoted as y, which represents probabilities of current input soccer player's highlight video belonging to the corresponding fine-grained style. Finally, the output y can be used to calculate cross-entropy loss for training the whole network. The cross-entropy loss, given each input data, is calculated as

$$\mathcal{L} = -\sum_{c=1}^{M} y_c log(y) \tag{2.8}$$

where M is number of classes, log is the natural log function, y_c is the binary indicator (0 or 1) if class label c is ground-truth or not and y is the predicted probability that the input is of class c. The architecture of energy-motion features aggregation module is shown in Fig. 2.1(c).

Figure 2.5: Examples from our fine-grained soccer players highlight video datasets. Fine-grained styles from the first row to the last row: Stepover, Elastico, Chop, Penalty-Kick shooting, Goal shooting and Free-Kick shooting.

2.4 Experimental Results

2.4.1 Soccer Players Highlight Video Datasets

We collect soccer players highlight video datasets from three different sources. Overall, there are two coarse-grained classes of soccer players actions, dribbling and shooting, and six fine-grained types of players actions. Fig. 2.5 presents examples of our dataset and Table 2.1 shows a summary of our dataset. We provide the details of data source, collection and statistics in following paragraphs.

Table 2.1: Summary of the Soccer Players Highlight Video Datasets. Abbreviations: P: Soccer Player, B: Soccer Ball, G: Goalkeeper, DW: "Defensive Wall", OoI: Object of Interest.

Soccer Players Actions		No.	Resolution	OoI
Dribbling	Stepover	102	min(720×1920)	P
	Elastico	49		
	Chop	82	max(1280×1920)	B
	Subtotal	233		
Shooting	Penalty-Kick	114	min(240×320)	P, B, G
	Goal	183		
	Free-Kick	39	max(720×1280)	P, B, G, DW
	Subtotal	336		P, B, G, DW
Total		569	Multiple resolutions	P, B, G, DW

Soccer Players Dribbling Videos: Dribbling is one of the most fundamental skills in soccer games. In particular, only these top-tier soccer players can perform professional offending dribbling skills smoothly, so we start to collect soccer dribbling videos from real soccer matches by searching and crawling on Youtube. We collect 233 soccer dribbling video clips from real soccer games with more than 4,600 frames and each dribbling video clip is annotated with the corresponding fine-grained dribbling style name: Stepover, Elastico and Chop. Dribbling styles annotations are based on the terminology used in soccer games. The **Stepover** is the style where soccer players will use their non-dominant foot to pretend kicking the ball to one direction but go over the ball in actual to evade defenders. The **Elastico** is the style where soccer players use outside of their dominant foot to push the ball to one direction, then change to move to reverse direction with ball. The **Chop** is the style where soccer players use one foot to kick the ball to the reverse direction behind their body. To elaborate, there are 102, 49 and 82 video clips of each dribbling styles Stepover, Elastico and Chop, respectively. It should be noted that these dribbling skills are always performed by soccer players within 1 second to ensure ball possession for subsequent movements. The average length of

video clip is around 0.9 seconds. The minimum length is only 0.5 seconds. The maximum length is 4.9 seconds. Most of video clips are at the resolution of 720×1920 with a few of the video clips are at 1280×1920. Each video clip contains soccer players and ball which are the objects of interests.

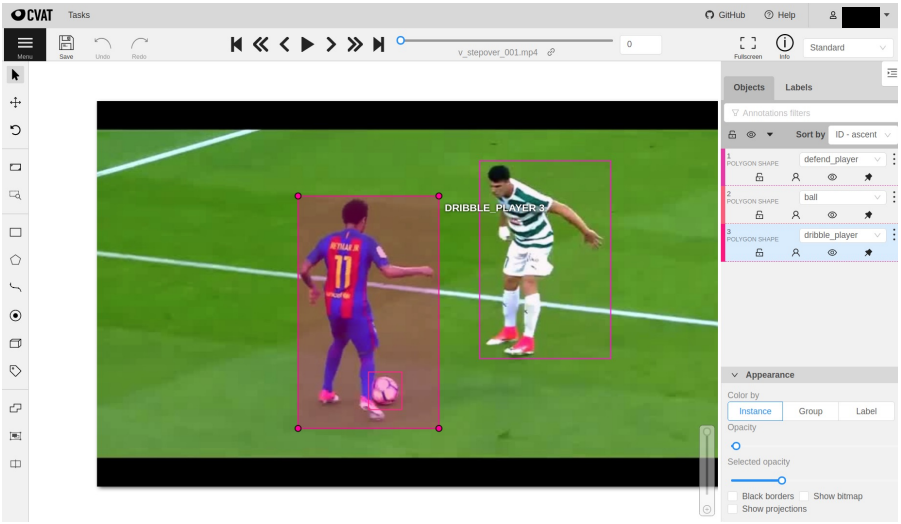

Figure 2.6: The view of CVAT and an example of annotating operation.

Soccer Players Shooting Videos: Soccer players shooting videos are collected from two sources in three different fine-grained type: Penalty-Kick Shooting, Goal Shooting and Free-Kick Shooting. The **Penalty-Kick shooting** is the case when a soccer player is allowed to take a single shot on the goal while it is defended seriously by the opposing team. The shot is taken from the penalty mark, which is 11m (12 yards) from the goal line and centred between the touch lines in soccer field. **Goal shooting** is the action when soccer players kick the soccer ball for goal. **Free-Kick shooting** is the case when an unopposed kick is taken by a player to restart the play after an opposition player has committed a foul. A player must take a free-kick from the exact location where the offense has occurred, and the play does not restart until the ball clearly moves.

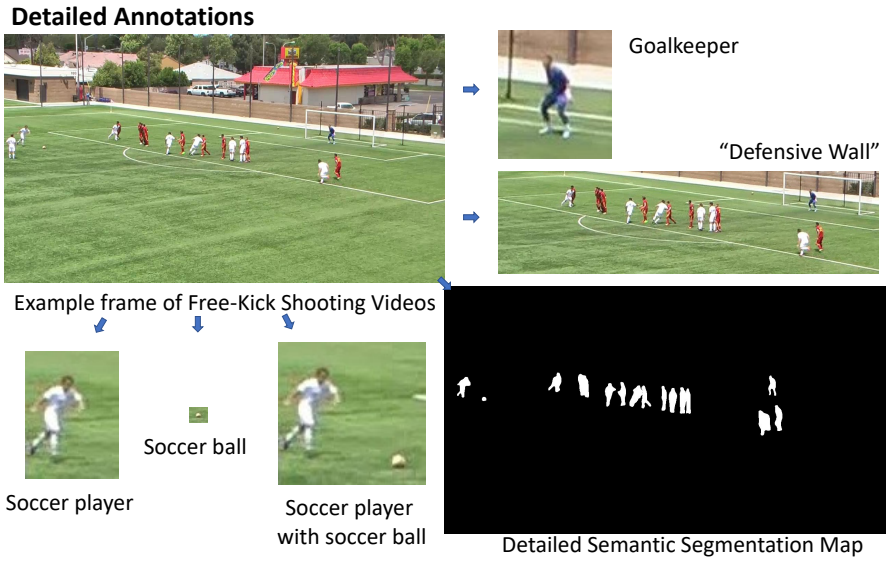

Figure 2.7: Detailed multi-level annotations of one example frame in a soccer player highlight video.

Firstly, as Penalty-Kick shooting happens rarely in soccer matches, we collect videos that belong to the class "SoccerPenalty" from the UCF-101 [384] dataset which contains videos from 101 different classes. In total, there are 114 videos with more than 11,000 frames, including soccer matches that are captured from different soccer leagues around the world and soccer training highlights played by teenage players. The resolution of all videos is 240×320.

Secondly, for the players Goal shooting and the Free-Kick shooting videos, we collected video data from three different soccer matches. The matches played by the teams were recorded using a single Canon XA10 video camera. The camera was installed at a height of 15 feet and 20 feet away from the horizontal baseline of the soccer field. The resolution of the recorded video is 1280×720. The camera operator was allowed to pan and zoom depending on where the action is happening on the soccer field in order to collect high resolution (720×1280) and good quality images with enough pixels on a player's body. For videos in these three matches, we cropped video

clips where soccer players are performing these two actions and collected the data as part of our soccer players highlight video datasets. There are 183 Goal shooting video clips with more than 23,000 frames and 39 Free-Kick shooting video clips with more than 4,700 frames. For all players' fine-grained shooting video clips, the average length is around 3.5 seconds. The minimum length is 2 seconds. The maximum length is 7.3 seconds.

Overall, our dataset is setup with the following features:

- Soccer players in our dataset are from a variety of soccer games, including soccer matches of top leagues in Europe, Asia, etc., as well as soccer matches among high school teams in the USA.
- Data are collected with diversity of participants maintained. Soccer players include professional adults players from various countries and students from high schools.
- Data are at varying resolutions. Most of the videos are at high resolution which is either 720×1920 or 720×1280.
- Detailed annotations are provided. Soccer players, ball, goalkeepers, etc., are annotated in each frame, which can be used for multiple purposes (See Section 2.4.2).

2.4.2 Annotations

We use the open-source annotation tool-CVAT [367] to annotate soccer-related objects of interests in each frame of soccer players highlight videos. CVAT provides users with a set of convenient instruments for annotating digital images and videos. It allows users to annotate images

with four types of shapes: boxes, polygons, polylines, and points. The collected dataset was annotated by five experts (initials of the annotators are: RL, TF, SW, XZ, AS).

Table 2.2: Annotation time of the the fine-grained action label on Soccer Players Highlight Video Datasets.

Soccer Videos	Soccer-related Actions	Annotating Time
Dribbling	Dribbling	5s∼10s
Dribbling	Defending	
Shooting	Shooting	20s∼30s
Shooting	Defending	
Shooting	Goal Saving	
Shooting	Defending with "Defensive wall"	

Firstly, we load each soccer player's highlight video on CVAT platform, given an image frame, we draw bounding boxes over the soccer-related objects of interests, *e.g.*, soccer dribbling players, soccer ball, goalkeepers, etc. Secondly, we go through all the frames in each video and annotate all soccer-related objects. Finally, we export the dataset and corresponding annotations in various formats, including COCO object detection [254], Pascal VOC semantic segmentation [118], etc., so that researchers can use the dataset for different purposes. An annotation example is shown in Fig. 2.6. It should be noted that soccer player's highlight video might contain non-relevant content, *e.g.*, soccer players are waiting for whistles from the referees. During the annotating process, we skip these frames. Therefore, in each highlight video clip, we concentrate on the period when the soccer players of interest are demonstrating professional skills and ensure the player's action is complete.

We also provide annotations of goalkeepers and players who are involved in the "defensive wall" which is the tactic that is particularly used in Free-Kick shooting. A detailed multi-level annotation example is shown in Fig. 2.7.

Table 2.3: Annotation time for the Soccer-related Objects in Soccer Players Highlight Video Datasets.

Soccer-related Objects of Interest	Bounding Box (per object)
Dribbling Players	$5s \sim 7s$
Shooting Players	$4s \sim 5s$
Goalkeepers	$4s \sim 5s$
Soccer Balls	$\sim 2s$
Defense Players	$4s \sim 7s$
Goal Area	$\sim 10s$
"Defensive Wall"	$5s \sim 10s$

In the following paragraphs, we present annotation efforts spent on the collected dataset. To record annotating time, we randomly select 10% of video clips for each fine-grained action based on the length of a video. Table 2.2 presents the annotation time for the fine-grained action label. Annotating shooting videos normally takes longer because shooting video clips always contain more soccer players and it takes longer to identify fine-grained actions played by soccer players and goalkeepers. Table 2.3 presents the annotation time to obtain the bounding box for each object of interest in each frame in soccer videos. Table 2.4 presents the runtime for detection, semantic segmentation and human body keypoints results for each frame in soccer videos by using Mask R-CNN [162] and Openpose [48].

Table 2.4: Runtime for generating semantic segmentation and human body keypoints by runing third-party algorithms on Soccer Players Highlight Video Datasets.

Algorithms	Outputs	Speed (FPS)
Mask R-CNN [162]	Detection, Semantic Segmentation	$\sim$8.5
Openpose [48]	Human body Keypoints	$\sim$10.0

Table 2.5: Results of soccer player and soccer ball detection with Mask-RCNN framework. Detection performance measured by average IOU (%) and inference speed (FPS) are shown.

Soccer Video Type	Average IOU (%)		Average Speed (FPS)
	Soccer Player	Soccer Ball	
Dribbling	84.48%	65.82%	8.43
Penalty-Kick	57.22%	26.49%	9.36
Goal	57.97%	23.87%	8.19
Free-Kick	56.01%	25.14%	8.00
Average	63.92%	35.33%	8.50

2.4.3 Implementation Details

We trained and evaluated our approach on the collected highlight video datasets. The framework of our approach is implemented using PyTorch [315] on 1 NVIDIA 1080Ti GPU.

2.4.4 Results of Soccer Player and Ball Detection

The first experiment evaluates the detection performance of soccer players and soccer balls. As introduced in Section 2.3.1, we use the Mask R-CNN [161] for soccer player and soccer ball detection because it can output bounding boxes, segmentation masks and human body keypoints in a single pass. We used Intersection Over Union (IOU) between the ground-truth and predicted bounding box and the speed of performance during inference in Frames Per Second (FPS) as metrics. The model is trained on the COCO dataset [254] and then evaluated using highlight videos of soccer players. We chose the model trained on the COCO dataset because it has a lot of diverse images for the class "Person" to which soccer players and goalkeepers belong to and the class "sportsball" to which class the soccer balls belong to. Table 2.5 shows the detection results on the soccer highlights videos. It should be noted that the detection results are based on all ground-truth annonations in the dataset, not simply the dribbling and shooting players. From Table 2.5 it

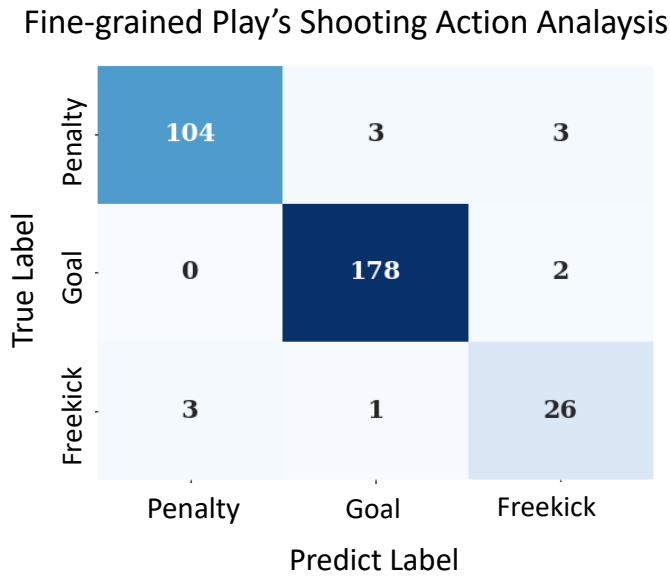

Figure 2.8: Confusion matrix of players' fine-grained shooting action analysis using our method.

can be observed that average IOU accuracy of player detection is 63.92% and soccer ball detection is 35.33%, with an average of 8.5 FPS inference time. It is to be noted that the average IOU accuracy of both soccer players and soccer balls in soccer dribbling videos are far higher than the IOU accuracy of that in shooting action videos, which is 84.48% for soccer players and 65.82% for soccer balls. This is because dribbling videos are captured using a camera that is moving along with soccer players when players are performing actions, while shooting action videos in our dataset are recorded using a camera with a larger field-of-view located far away from soccer players when they perform these actions. This setting is common in soccer games because for dribbling, people are more focused on how soccer players are performing actions; while for shooting, people are also looking at referees, goalkeepers who are always at a distance between soccer players.

2.4.5 Results and Comparisons of Soccer Players' Fine-grained Action Analysis

We present results and discussions on shooting, dribbling and mixed shooting and dribbling analysis, respectively. We compare the proposed method with a range of existing algorithms which belong to three categories. The *first* category includes video-based methods. To elaborate, we use 3D-ResNet with two network designs (18 and 50 layers) proposed by Kataoka *et al.* [158, 203] and ResNet(2+1)D proposed by Tran *et al.* [403]. In 3D-ResNet, full 3D convolution is carried out while in ResNet(2+1)D networks, a spatial 2D convolution followed by a temporal 1D convolution is used in each of the layers. Both works are designed for activity recognition and their models are pre-trained on Kinetics [50] dataset. We mainly fine-tune the last convolutional blocks as well as final fully-connected layer on soccer highlight video datasets. Besides, we also test the video swin transfomer [276] by fine-tuning last two convolutional blocks with the Kinetics [50] pretrained model on our datset. The *second category* contains skeleton-based methods. We use ST-GCN [451], ST-RT [325] and PoseC3D [112]. The *third category* mainly contains the baseline method which is presented in [242]. On each task, we split soccer highlight videos into train-validation-test sets. For the dribbling videos, as they are collected from a wide range of Youtube videos from diverse sources, we perform random sampling to generate the the split. For the shooting videos, as the match ids of partial dataset are available, we perform random sampling per match to generate split if the match id is available. We evaluate different algorithms using 5-fold cross validation for a fair comparison and present average precision, recall and F1-score [142] for all players actions. The average precision (AP) is defined as: AP $= \frac{1}{N}\sum_{i=1}^{N} Precision_(i)$, the average recall (AR) is defined

Table 2.6: Results of players' fine-grained shooting types classification on soccer players shooting highlight videos. Best results are in bold. Players' fine-grained shooting actions: Penalty-Kick, Goal, Free-Kick. 18 & 50: 3DResNet with 18 and 50 layers.

Methods	Registration	AP	AR	A-F1
Tran *et al.* [403]	✗	0.480	0.492	0.486
Kataoka *et al.* [203] (18)	✗	0.672	0.649	0.660
Kataoka. *et al.* [203] (50)	✗	0.591	0.564	0.578
Liu. *et al.* [276]	✗	0.717	0.65	0.681
Yan *et al.* [451]	✗	0.665	0.646	0.656
Plizzaria *et al.* [325]	✗	0.835	0.671	0.741
Duan *et al.* [112]	✗	0.573	0.554	0.563
Li *et al.* [242] (baseline)	✓	0.852	0.745	0.788
EMA-Net (Ours)	✓	**0.929**	**0.933**	**0.931**

as: AR $= \frac{1}{N}\sum_{i=1}^{N} Recall_(i)$ and the average F1-score (A-F1) is defined as: A–F1 $= \frac{1}{N}\sum_{i=1}^{N} F1$ $–score_(i)$, where in each equation, i is the each action class, N is the number of total action classes.

Comparisons and Results of Players' Fine-grained Shooting Classification

Soccer players' fine-grained shooting evaluation is conducted with 242 training 30 validation and 64 test video clips. Table 2.6 presents the results and comparisons of players' fine-grained shooting action analysis between our method (***EMA-Net***) and a range of video-based algorithms. From Table 2.6, it can be observed that our method (***EMA-Net***) achieves the best performance in classifying players' fine-grained shooting actions, with the average f1-score of 0.931, which is $\sim$0.15 higher compared with the baseline method proposed by Li *et al.* [242]. We note that more gain comes from the high average recall, which is 0.933 using our method. It is observed that video-based methods which use standalone convolutional neural networks achieve relatively low performance in players' fine-grained action classification. For instance, the best performance is obtained with as average f1-score of 0.66 by using 3D-ResNet18. Video swin transformer shows

good generalizability on fine-grained action analysis compared with 3D-ResNet, which gives 0.68 as the average f1-score. Among skeleton-based human action analysis methods, ST-RT proposed by Plizzaria *et al.* [325] gives the best performance with 0.741 as the average of f1-score. We discuss that these methods mainly show advantages on coarse-grained action analysis driven by large-scale training strategy, which is extremely challenging in fine-grained analysis with limited data and annotations, especially in sports community. However, using energy image representation can leverage complex temporal and spatial information into one single image representation for the model to learn fine-grained features for players action classification; this is validated by comparing the results with Li *et al.* [242].

Our method, which takes the next subsequent step to aggregate explicit motion features from the motion modelling module with energy features extracted from energy image representation provides the best performance in classifying players' fine-grained shooting actions. It also validates our expectations that by using motion features as an auxiliary feature set aggregated with energy features, the model can perform better with the aggregated features due to the capture of enhanced motion dynamics. Besides, it can be observed that image registration is important for players' fine-grained action classification. This is because without players registration, the motion of the players across the video sequences and the motion of each part of human body within each frame are quite different and this causes feature-points-based transformation in standalone convolutional neural networks to be inaccurate. Therefore, our framework not only extracts features from registered energy image representation for players actions but also extracts features by encoding motion dynamics of soccer players explicitly, and both modules collaboratively provide improved model performance.

Fig. 2.8 presents the confusions matrix of players' fine-grained shooting action analysis using our method. It can be observed that penalty shooting style is easier to be miss-classified to other two styles, especially to free-kick shooting. Our method shows good performance in classifying goal shooting. However, the current dataset still suffers from insufficient data, especially for free-kick shooting. We expect to continue extending the current dataset and provide extensive discussions in Section 2.4.7.

Comparisons and Results of Players' Fine-grained Dribbling Classification

Soccer players' fine-grained dribbling evaluation is conducted with 157 training, 31 validation and 45 test video clips. Table 2.7 presents the results and comparisons between our method (***EMA-Net***) and a range of existing video-based methods. From Table 2.7, it can be observed that our method (***EMA-Net***) achieves the best performance, with the average 0.767 f1-score, which is around $\sim$0.09 higher compared with the baseline method proposed by Li *et al.* [242]. Moreover, similar observations, with those in Sec. 2.4.5, are made that methods with standalone 3d convolutional neural networks cannot give ideal performance for classifying fine-grained actions. One of the reasons is that training such kind of models requires a large amount of data, but only few soccer video datasets contain dribbling actions so obtaining a large number of video data for training is expensive. Video swin transformer has shown improved performance with the average f1-score of 0.632. Among skeleton-based methods, the method proposed by Duan *et al.* [112] can achieve 0.624 on average f1-score. For fine-grained action analysis, our method, which uses energy image representation can leverage complex motion and spatial information into one single image representation which provides strong and valid signals for the model to learn fine-grained features for players action classification. In addition, our method fully exploits and encodes motion dynamics in

Table 2.7: Results of players' fine-grained dribbling styles classification on soccer dribbling highlight videos. Best results are in bold. Fine-grained dribbling actions: Stepover, Elastico, Chop. 18: 3DResNet with 18. †: only parameters of the last block in the model are optimized.

Methods	Registration	AP	AR	A-F1
Kataoka. *et al.* [203] (18)	✗	0.525	0.439	0.478
Kataoka. *et al.* [203]† (18)	✗	0.619	0.449	0.520
Liu. *et al.* [276]	✗	0.656	0.610	0.632
Yan *et al.* [451]	✗	0.553	0.517	0.534
Plizzaria *et al.* [325]	✗	0.661	0.546	0.598
Duan *et al.* [112]	✗	0.718	0.553	0.624
Li *et al.* [242] (baseline)	✓	0.674	0.668	0.671
EMA-Net(Ours)	✓	**0.776**	**0.760**	**0.767**

motion features which are aggregated with energy features. This enables the model to further learn fine-grained signals for improved fine-grained dribbling action classification performance.

Fig. 2.9 presents the confusions matrix of players' fine-grained dribbling action analysis using our method. It can be observed that the stepover style is easier to be miss-classified to chop style. A relative good performance can be observed in classifying stepover style.

Compared with the player's shooting actions, we argue that player's dribbling analysis suffers from poor performance due to four main reasons. First, dribbling skills are way more complicated than shooting styles. Based on our empirical observations, only top groups of professional players can consistently show fancy dribbling skills in professional soccer leagues where soccer freshmen and fans always try to imitate these skills. Second, the status of a soccer player who is performing dribbling skills is quite different from the one in performing shooting. Soccer players are moving at a fast speed when they are performing dribbling skills and sometimes there are physical confrontations. While in shooting movements, especially, in penalty shooting and free-kick shooting, soccer players are moving in a relatively stable conditions and are supposed to be able to

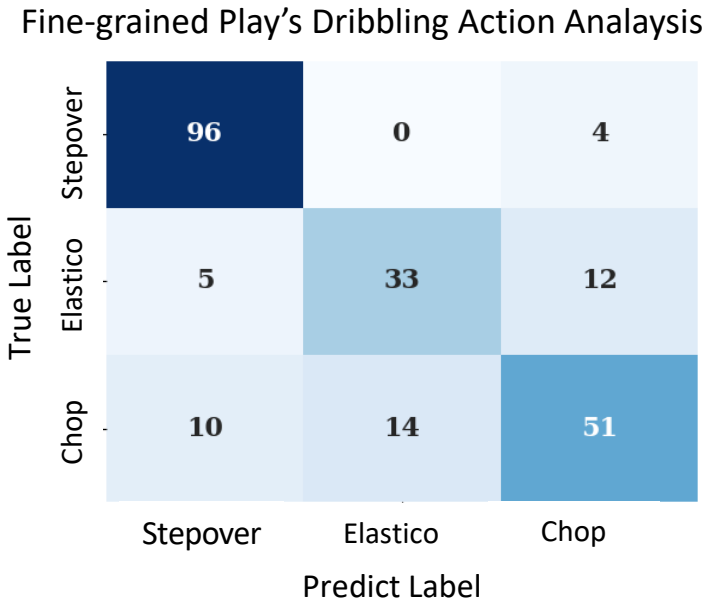

Figure 2.9: Confusion matrix of players' fine-grained dribbling action analysis of our method.

perform shooting movements much more smoothly, and this can largely remove noises from fine-grained shooting analysis. Third, it should be noted that our paper explicitly includes the soccer ball in all players' fine-grained analysis. Soccer ball moves differently and it can implicitly provide cues for helping players' fine-grained shooting analysis. Fourth, we only concentrate on analyzing the fine-grained movements performed by the dribbling player and soccer ball in dribbling analysis. While, additional contextual information is leveraged in shooting analysis, which is explained as one limitation in Section 2.4.7.

Comparison of Results of Players' Fine-grained Action Classification

Soccer players' fine-grained shooting and dribbling evaluation is conducted with 397 training, 62 validation and 110 testing video clips. Experimental results are shown in Table 2.8. From Table 2.8, we observe that our method (***EMA-Net***) achieves the best performance in classi-

fying fine-grained actions of various styles, with the average 0.794 f1-score, which is 0.09 higher compared with the method proposed by Li *et al.* [242]. We can also observe that video swin transformer gives competitive performance with the average 0.684 f1-score, which demonstrates the superiority of transformers networks. Among skeleton-based methods, the method proposed by Duan *et al.* [112] achieves the average 0.671 f1-score, which shows its robustness when more data are given for training. Our method, which uses energy features from energy image representations and motion features from segmentation maps can collaboratively provide strong signals for the model to learn fine-grained features for action classification.

Table 2.8: Results of players' fine-grained actions classification including dribbling and shooting actions on soccer players highlight videos. Best results are in bold. Players' fine-grained actions: Stepover, Elastico, Chop, Penalty-Kick, Goal, Free-Kick. 18 & 50: 3DResNet with 18 and 50 layers.

Methods	Registration	AP	AR	A-F1
Kataoka. *et al.* [203] (18)	✗	0.502	0.541	0.521
Kataoka. *et al.* [203] (50)	✗	0.672	0.476	0.557
Liu. *et al.* [276]	✗	0.717	0.654	0.684
Yan *et al.* [451]	✗	0.619	0.624	0.621
Plizzaria *et al.* [325]	✗	0.528	0.570	0.548
Duan *et al.* [112]	✗	0.707	0.639	0.671
Li *et al.* [242] (baseline)	✓	0.713	0.698	0.705
EMA-Net (Ours)	✓	**0.799**	**0.790**	**0.794**

Fig. 2.10 presents the confusions matrix of players' fine-grained shooting and dribbling action analysis using our method. We observe a decrease in classification performance on dribbling styles, while classification performance on shooting styles remains relatively good. In particular, goal shooting has become one of the major classes that causes confusion. We argue that the main reason for it is that the goal shooting styles are performed under similar conditions as dribbling movements, *e.g.*, high speed, physical confrontations. To mitigate this issue, we discuss that adding

an umbrella classification module, for close view or long view, could be helpful in improving soccer players' fine-grained analysis with all fine-grained styles. However, we argue that it could pose additional problems. In players' shooting movements, we have observed that the camera is always moving from long view to close view because the camera is controlled to capture the scene with large field-of-view at the beginning and then the camera is steered to move in a fast speed to track the soccer ball, which might cause difficulties in distinguishing view types (narrow-field-of-view ("close") vs. wide-field-of-view ("far/long")) and it could make simply estimating long view and close view not work. Based on this observation, we discuss one potential solution which is explained in the following. One can add an additional branch, named as view estimation, along with player detection and keypoints parsing modules to estimate view corresponding to each frame in a video clip. Then the estimated view information can be encoded in the motion modelling module or a separate light module can be designed to output view elapsing features. We believe it could be helpful but this requires additional camera meta information, which could be an interesting future work.

2.4.6 Ablation Study

In this section, firstly we perform ablation studies to evaluate how features from each module contributes to the proposed framework (***EMA-Net***). Secondly, we conduct extensive experiments on players' fine-grained shooting action classification with the consideration of goalkeepers actions in soccer games.

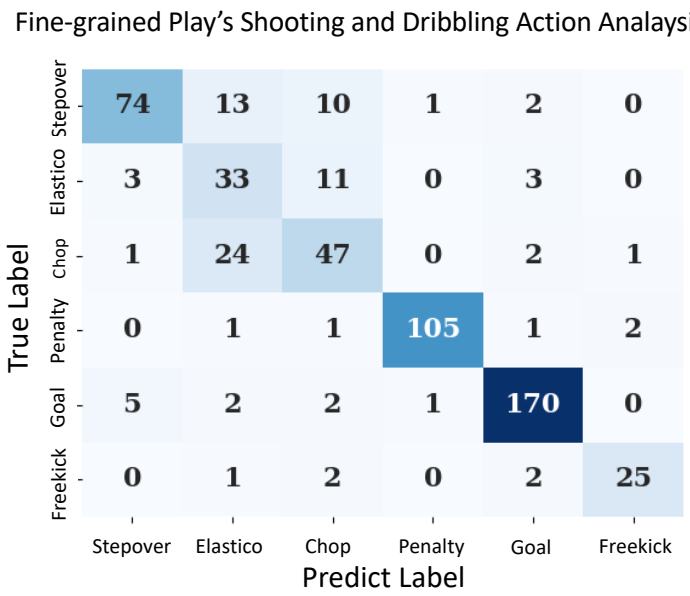

Figure 2.10: Confusion matrix of players' fine-grained shooting and dribbling action analysis.

Table 2.9: Ablation Results of players' fine-grained shooting action classification on player shooting highlight videos. Best results are shown in bold. Players' fine-grained shooting actions: Penalty-Kick, Goal, Free-Kick.

Methods	Energy Features	Motion Features	Fusion Method	A-F1
EMA-Net	✓	✗	✗	0.788
EMA-Net	✗	✓	✗	0.424
EMA-Net (LSTM Motion)	✓	✓	Naive Fusion	0.817
EMA-Net (GRU Motion)	✓	✓	Naive Fusion	0.886
EMA-Net **(Ours full model)**	✓	✓	Naive Fusion	0.910
EMA-Net (Ours full model)	✓	✓	Energy-Motion Aggregation	**0.931**

Contributions of Features from Each Component

We run ***EMA-Net*** with three different network designs, which are with energy features module removed, motion modelling module removed and with none of the modules removed. Be-

sides, we also conduct ablation experiments by using LSTM and GRU functions for designing motion modelling module and simply using a naive feature fusion module to replace the proposed energy-motion features aggregation module. The tested naive feature fusion module consists of a series of fully-connected layers with normalization, activations and DropOut layers. We conduct experiments on soccer players' fine-grained shooting highlight videos and the experimental results are shown in Table 2.9. From Table 2.9, the ***last*** row shows the results by using the ***EMA-Net*** which is the proposed framework. The ***third*** row is the results by using the ***EMA-Net*** without energy features. The ***second*** row shows the results by using only the energy features, which is also the same as the baseline method proposed by Li *et al.* [242]. It can be observed that using only energy features extracted from energy image representation gives better performance (0.788 vs. 0.424) than using only the motion features from motion modelling module. Using motion features alone would saturate the model to learn valid signals for players' fine-grained action classification, which is also observed in Sec. 2.4.5 and Sec. 2.4.5. However, using motion features as an auxiliary signal to be aggregated with energy features can improve the model performance significantly. In addition, from the ***fourth***, ***fifth*** and the ***last*** rows in Table 2.9, it is observed that motion features outputted from the motion modelling module with different designs can consistently help in players' fine-grained action analysis. Among various motion modelling designs, our method with transformer-based design gives the best performance. Finally, by comparing the ***last two*** rows, it is observed that, compared with a naive feature fusion method, the proposed energy-motion features aggregation module shows superiority in aggregating both energy features and motion motion features for soccer players' fine-grained action analysis.

Table 2.10: Ablation results of fine-grained action classification on soccer players shooting videos with the considerations of goalkeepers' actions. Players' fine-grained actions: Penalty-Kick, Goal, Free-Kick, Goalkeepers' Goal Saving.

Methods	A-F1
Li *et al.* [242] (Baseline)	0.563
EMA-Net (Ours)	**0.683**

The Role of Goalkeeper's Saving Actions

As far as we know, none of the existing work has conducted action analysis that involves both soccer players and goalkeepers. We conduct players action classification by considering actions performed by soccer players for Penalty-Kick shooting, Goal shooting and Free-Kick shooting, and Goal saving by goalkeepers. We use all 336 players shooting videos and split them into train-validation-test sets where each consists of 241, 30 and 65 video clips, respectively. For each video clip, we use goalkeeper tracklet separately, resulting in additional 230 tracklets for training, 30 tracklets for validation and 65 tracklets for testing. Table 2.10 presents ablation experimental results for 5-fold cross validation and comparison with the baseline method proposed by Li *et al.* [242]. It can be observed that the proposed framework consistently outperforms the baseline method in classifying players' fine-grained actions in considering actions performed by soccer players and goalkeepers in soccer games.

2.4.7 Discussions on the Future Applications and Limitations

Compared with most of the existing video datasets, sports videos contain far richer information and are way complex in scenes, objects, motions, noises, etc., which pose sports video analysis to be extremely challenging. In this work, we focus on soccer players' fine-grained action analysis. We collect soccer highlight video datasets and provide annotations for each frame

in each video. This dataset can be extended by introducing a wider range of soccer players' fine-grained movements. For instance, one can analyze short ball pass and long ball pass, and passing soccer ball with different kicking styles, *e.g.*, back heel, side and instep, etc. Besides, we argue that "goalkeeper" plays an important role in soccer games and it has not received sufficient attention. Analyzing goalkeepers' actions could lead to important future directions. For instance, one can analyze the success/fail ratio of the "goal saving" performed by goalkeepers and such kind of analysis could be conducted given the data that includes the front view of goal area and goalkeepers' movements towards soccer ball. In addition, it would be interesting to perform fine-grained interaction analysis among a group of soccer players and the soccer ball, which could be beneficial to understand team tactics. Finally, researchers would also be interested in non-players' fine-grained action analysis, including the chair referee, two assistant referees, the fourth official referee, coaches and the ball boy, etc., and semantic analysis based on these identities could be conducted in the future.

Moreover, when extending soccer players action analysis from dribbling/shooting to other fancy actions, our joints-based image registration algorithm can be applied directly or extended accordingly based on certain fine-grained analysis of interests. For instance, to researcher who may be interested in players' head jump, joints of upper human body would become more interesting. Therefore, hip-joints based registration can be adapted to shoulder-joints based registration because we think that head poses are directly connected with the player's head movements and in turn support head motions for player's fine-grained head-jump analysis. Similarly, another example for throw-in can be explained as well. Joints-based registration can be utilized for analyzing long throw-in and short throw-in actions performed by players in soccer games. It is expected that extensive fine-grained actions and annotations to be consistently added in the future. After extending the

dataset, a wide-variety of objects, motions and tactics could be analyzed further, *e.g.*, fine-grained defending skills, referee hand gestures, coaches' tactics, etc. Besides, based on the extending dataset which is relatively large-scale, existing algorithms [112, 276] could be leveraged during the training and integrating these design ideas with current framework for new mechanism.

Chapter 3

RECLIP: Resource-efficient CLIP by Training with Small Images

3.1 Introduction

Representation learning is a foundational problem in computer vision and machine intelligence. Effective image representation can benefit a myriad of downstream tasks, including but not limited to image classification, object detection, semantic segmentation, and 3D scene understanding. In the past decade, the community has witnessed the rise of supervised learning [94, 385], then self-supervised learning [65, 160, 17], and most recently language-supervised learning [336, 188, 457]. Language-supervised representation gains much traction for its exceptional versatility. It exhibits outstanding performance in zero-shot classification [336], linear probing [336, 457], few-shot learning [496], full finetuning [108], and finds great applications in text-guided image generation [338]. Much like the role of supervised pretraining [94] before, language-

supervised pretraining has emerged as a simple yet powerful methodology for representation learning today.

Traditional supervised learning uses a predetermined set of labels, and is effective across a wide range of data and computational resources. In contrast, natural language offers richer learning signals such as object categories or instances, named-entities, descriptions, actions, and their relations at multiple levels of granularity. Unfortunately, this rich supervision also leads to a higher level of noise in the data, where many image-text pairs have only loose connections. To address this noise, data and computational scaling have proven to be highly effective and necessary. For example, training CLIP models require $\sim$3k V100-GPU-days, and likewise CoCa requires $\sim$23k TPU-v4-core-days. Apart from the lengthy training time, the large batch requirement of contrastive learning recipes also demand substantial amount of device memory at all times. These factors limit the research of language supervised learning to institutions with high-end infrastructure, and hinder the exploration by the broader community.

Thus, improving efficiency of contrastive training has drawn substantial research interest. For example, [468] precomputes the image features by a pretrained classification model to reduce the training cost. [467] utilizes sigmoid loss to avoid the use of all-gather operation and improves learning with a smaller batch size. Moreover, [454] leverages masked images to speed up contrastive learning. The community have also explored smaller batch sizes [109] or curated academic datasets [246, 231] for contrastive learning. However, it is not clear how well the findings in smaller batch and data size settings generalize to larger batch and data size.

We present RECLIP (Resource-efficient CLIP), a simple method designed to make CLIP more affordable and reproducible for the community (see Fig. 6.3). Consider images 1-3 in the top

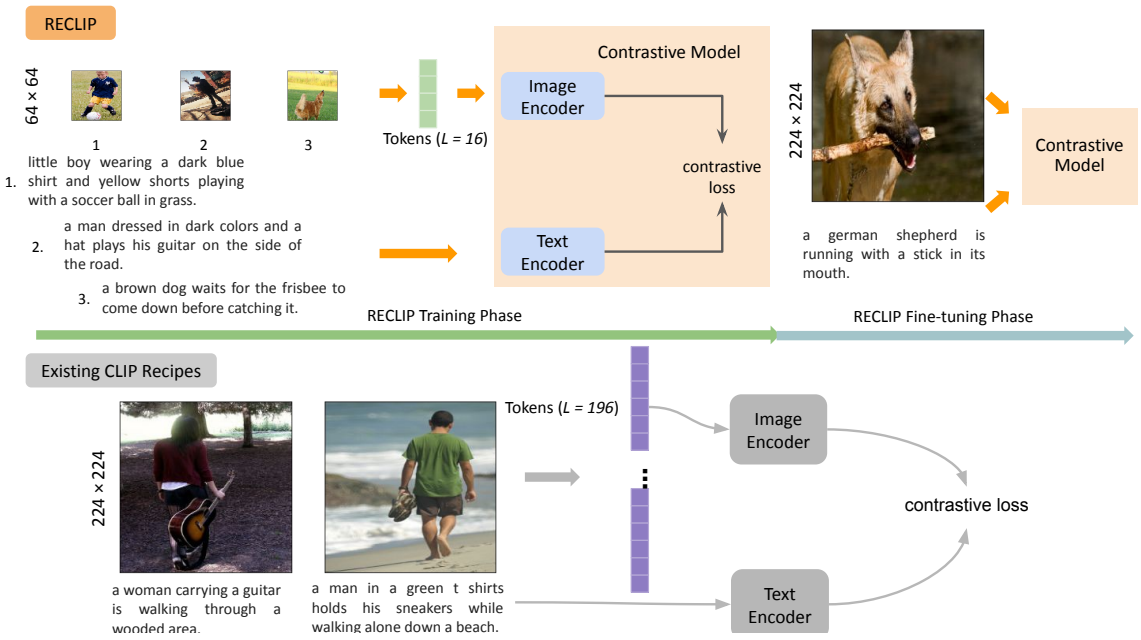

Figure 3.1: Top: Resource-efficient CLIP (RECLIP) training pipeline. Bottom: existing CLIP training methods. RECLIP leverages small images for the main training phase which significantly reduces computational resource requirements through much shorter image sequence length.

left of Fig. 3.1. Humans can effortlessly match the images with the corresponding texts below them, *e.g.*"a boy is playing a soccer ball in grass" matching image 1. Although the images are only of size 64×64, they contain adequate amount of visual information for pairing with texts. Our main insight is to train on small images during the main training phase, and finetune the model with high-resolution images for a short schedule in the end. Intuitively speaking, our approach re-introduces the idea of "coarse-to-fine" from classical computer vision to contrastive learning, whereby pretraining incorporates high-level information from small images and finetuning enables the model to refocus its attention on the important details. There is no need for multi-view supervisions [246, 454], feature distillation [231], other contrastive losses [467], pretrained classifiers [468], or image masking [247]. Surprisingly, RECLIP achieves highly competitive zero-shot classification and retrieval performance using 64×64 images, which significantly reduces computational resource usage. We

attribute this to the complexity of image tower being quartic with respect to the image size (see Eqn. 3.4).

In addition, RECLIP demonstrates the efficiency and effectiveness of using short sequence length for image language representation learning. Existing image-text pretraining methods typically use long sequence lengths, *e.g.*441 [336] or 784 [457] to achieve strong downstream zero-shot transfers. Long sequence image encoding has been validated to benefit image classification [27] and object detection [66] with vision transformers. [175] find the sequence length is a key factor for masked image representation learning. Different from these methods that advocate for long sequence length, RECLIP demonstrates that using only **16** tokens for the image encoding is sufficient for the main training phase, and can achieve highly competitive zero-shot transfer capabilities via a short high-resolution finetuning schedule. Interestingly, our image sequence length is 4 to 5$\times$ shorter than the *text* sequence lengths of popular recipes *e.g.*76 [336] or 64 [457].

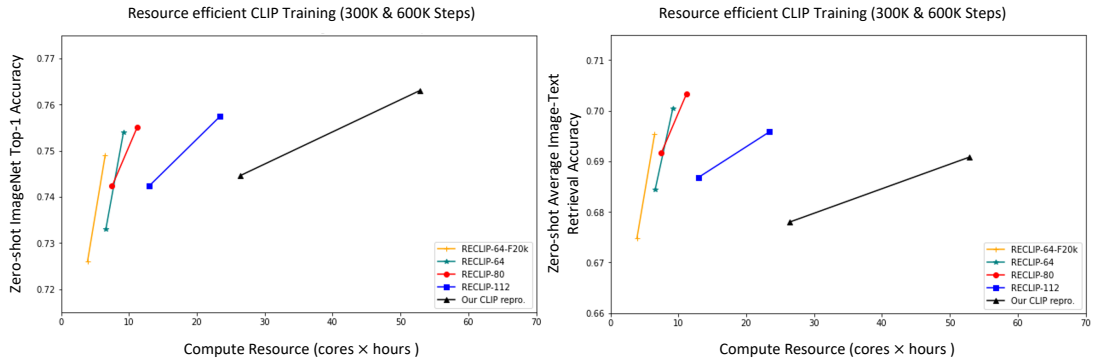

Figure 3.2: Zero-shot accuracy *vs.* compute resource in cores$\times$hours trade-off. RECLIP-X: RECLIP training for 300k and 600k steps with image size X where $X = 64, 80, 112$. RECLIP-64-F20k: RECLIP-64 finetuned for 20k steps. Our CLIP repro.: our reproduction of CLIP [336]. Zero-shot image-text retrieval results are averaged from image-to-text and text-to-image Recall@1 on two benchmark datasets, Flickr30K [327] and MSCOCO [70]. RECLIP consumes significantly less compute resource and is more accurate on zero-shot image-text retrieval and highly competitive classification results on ImageNet-1K validation set.

In Fig. 3.2, we present zero-shot classification and retrieval performance, and resource costs in cores$\times$hours of training RECLIP models and the baseline model for short and long schedules. Experiments show that, using the same batch size and training steps, RECLIP reduces the computation resources by 6 to 8$\times$ and largely preserves the classification and retrieval accuracy. When comparing to state-of-the-art (SOTA) methods, RECLIP significantly saves resource usage by 5 to 59$\times$ and shows highly competitive zero-shot classification and retrieval accuracy. Apart from image-level tasks, we explore transfer learning of RECLIP to open-vocabulary detection tasks [148], which typically requires high-resolution images for small object recognition. Surprisingly, RECLIP achieves 32 AP_r, matching the state of the art performance of RO-ViT [207] on LVIS benchmark. This demonstrates the potential of RECLIP for region and pixel-level tasks beyond image-level understanding. We believe RECLIP could enable the broader research community to explore and understand language supervised pretraining in a more resource friendly setting.

3.2 Related Work and Our Contributions

3.2.1 Learning with Low-Resolution Images

Deep learning techniques have been utilized on a wide-variety of computer vision tasks, *e.g.*visual recognition [162, 110], video analysis [403], images generations [338], etc. Most of existing work follow the standard training and testing paradigms to exploit very deep models by using images with the fixed resolution, *e.g.*224×224. This setting has been one of fundamental standards for various computer vision tasks. However, an increasing number of studies have been conducted to investigate to train deep learning models with low-resolution data. [402] have observed significant discrepancy on image sizes caused by augmentation methods during the train and test

period, and further validated the effectiveness of using lower resolution images for training than testing. Driven by the needs for specific tasks, *e.g.*face recognition, surveillance images analysis, etc., [379, 380] and [180] study learning with low resolution images and generally focus on using high resolution images as auxiliary data to help to train models with low resolution data, which causes difficulties to generalize on broader visual recognition tasks. For video understanding, [440] propose to use variable mini-batch shapes with different spatial-temporal resolutions for training deep video models and obtain optimal performance and time trade-offs. With recent advances of vision transformers [110, 159], [151] speedup image pretraining by using masked image modelling with low resolution data. [272] introduce a a log-spaced continuous position bias for pretraining vision models by using smaller images and transfer to high-resolution localization tasks.

3.2.2 Language-supervised Learning

Due to the natural co-occurrence of image and language data on the web, language-supervised learning has become a highly effective and scalable representation learning methodology. Researchers have explored a variety of paired image-text data such as image tags [71, 105, 192], captions [97, 357, 421, 370], alt-texts [188, 365], image search queries [336], page title [69], or a combination of these sources [69]. From a modeling perspective, contrastive learning is particularly suitable for recognition and retrieval tasks, because of its simplicity and versatility. However, the high requirements of computational resources have limited the research from the broader community.

To fully leverage capabilities of vision and language pretraining, large batch size (*e.g.*16k [188], 32k [336, 454], or 64k [457]) and web image text data have been adopted widely. This requires a large amount of computational resources which many academic institutions and industry labs can-

not afford. To address such limitation, [468] proposes to precompute the image features with frozen classifier backbone, while [467] proposes sigmoid loss which better supports small batch training. In addition, masked image learning [454], multi-views data augmentations [246, 454], knowledge distillations [231] and masked self-distillation [109] have been proposed. Since many of these methods are trained and evaluated on smaller scale/data, it is unclear how well they may scale up to larger batch and data. For example, [438] shows that the advantage of contrastive learning approaches on smaller scales may not always hold at larger scales. In contrast, we propose a simple and novel recipe for language image pretraining, which (1) significantly reduces computation resource requirements and (2) works well on a large-scale web dataset [69] with minimal change to the established CLIP recipe [336].

3.2.3 Contributions of this Chapter

- We present a new language image pretraining methodology, Resource-efficient CLIP (RECLIP) to minimize computational resource requirements.
- We leverage small images for the main contrastive learning phase to enable the model to be trained with language supervisions fast and then finetune the model on high-resolution data with a short schedule in the end.
- RECLIP significantly saves compute resource, reduces FLOPs and achieves highly competitive performance on both zero-shot classification and image-text retrieval benchmarks.
- RECLIP matches the state of the art in open-vocabulary detection with much less training resources.

3.3 Technical Approach

3.3.1 Preliminaries

Contrastive Language Image Pretraining. Following existing works [336, 457], we utilize a transformer-based contrastive model which consists of an image encoder and a text encoder. The image and text encoders are trained to output image-level representation and sentence-level representations respectively. The image embeddings $\{p\}$ and text embeddings $\{q\}$ are obtained by global average pooling at the last layers of image and text encoders. The cosine similarity of the embeddings in batch B, scaled by a learnable temperature τ are the input to the InfoNCE loss [309, 336]. The image and text contrastive loss is obtained by $L_{con} = (L_{\text{I2T}} + L_{\text{T2I}})/2$, with:

$$L_{\text{I2T}} = -\frac{1}{B}\sum_{i=1}^{B}\log(\frac{\exp(p_i q_i/\tau)}{\sum_{j=1}^{B}\exp(p_i q_j/\tau)}). \tag{3.1}$$

$$L_{\text{T2I}} = -\frac{1}{B}\sum_{i=1}^{B}\log(\frac{\exp(q_i p_i/\tau)}{\sum_{j=1}^{B}\exp(q_i p_j/\tau)}). \tag{3.2}$$

where i, j are indexes within the batch. This loss is optimized to learn both the image and language representation in the dual-encoder model.

3.3.2 Resource-efficient CLIP

At a high-level, our method utilizes small images to reduce computation and leverage a brief finetuning stage at the end of training to adapt for high-resolution inference. Intuitively, the use of smaller images presents a trade-off between how much detail we encode per example and how many samples we process per unit of computation resource. Fig. 6.3 shows the RECLIP training pipeline on the top. There are two phases: low-resolution main training, and high-resolution finetuning. In the first phase, we leverage small images which contain sufficient visual concepts with

paired texts as the input to the image and text encoders. By using an image size of 64 and a text length of 16, RECLIP processes the training data significantly faster than existing methods. In the second phase, we finetune the model for a short cycle on high-resolution data to provide valuable image details, which largely enhances the representation quality of the model. Below we delve deeper into specific aspects of RECLIP design.

Structure preservation by learning from small images. In Fig. 6.3, we observe that small images can preserve visual structure and contain sufficient concepts well. For instance, human can easily tell the object, "a dog", in the third image and associate the image with the text of "a brown dog waits for ...", and this is a fundamental principle for our RECLIP to leverage small images for the main language-supervised pretraining. Because down-sampling is a structure-preserving operation *i.e.*global appearance remains similar, we are able to reduce the token length aggressively without compromising the performance of the model. This is different from other techniques to reduce the sequence length (*e.g.*random masking) where the global appearance may change significantly with reduced sequence lengths. Additional visualization presents a comparison between various image resolutions and sheds light on how small images effectively preserve visual appearance (see Fig. 3.3).

Training complexity with small images. The computation cost of contrastive learning mostly depends on the cost of processing images [336, 247, 457], partly because the image encoder is typically heavier than the text encoder, partly because the image token length tends to be greater than that of text tokens. Below we provide theoretical analysis to understand the efficiency of using small images.

Let the number of tokens from the image encoder be:

$$N = hw/p^2 \tag{3.3}$$

, where h/w are height/widths of the image, and p is the patch size. If we replace h by H/r and w by W/r, where H/W are the original image height and widths, and r is the down-sampling factor. The computation complexity C of the image encoder of a batch is given by:

$$C = O(BN^2) = O(\frac{BH^2W^2}{p^2r^4}) \tag{3.4}$$

, where B is the batch size. When B, H, W, p are held constant, we have:

$$C = O(\frac{1}{r^4}) \tag{3.5}$$

This shows that reducing the image size is very effective in reducing computation complexity to the inverse power of up to 4. Since image encoder is the computation bottleneck in existing CLIP recipes [336, 247, 468, 457], RECLIP reduces the image sequence length to **16** by using an image size of 64, which makes our image token length the same as our own text token length, and much shorter than those of aforementioned methods.

The above complexity analysis C is calculated based on the core operations self-attention layers in transformers. However, empirically the complexity of a transformer may not be dominated by the self-attention layers, the fully connected layers also play an important role. GPT-3 [39] paper have provided computation analysis of their language models, where the computation cost is estimated as $O(N)$, linear with the sequence length. Thus, we discussed the lower-bound of the complexity C_{lb} of RECLIP in (3.6). Using the notation of (3.4) and (3.5), we have

$$C_{lb} = O(BN) = O(\frac{BHW}{pr^2}), \tag{3.6}$$

During the training, the B, H, W, p are normally constant, so (3.6) can be simplified as:

$$C = O(\frac{1}{r^2}). \tag{3.7}$$

Compared to (3.5), we observe that the computation savings in practice may be somewhere between $O(\frac{1}{r^2})$ and $O(\frac{1}{r^4})$. This analysis shows that changing r is very effective regardless of the compute estimation techniques.

Constant batch size. Batch size is a critical factor in contrastive learning [336, 319, 247, 66] and larger batch has consistently yielded improvement. In Equation 3.4, the complexity changes linearly with batch size B. Observing that reduced batch size tends to hurt representation quality, we keep the batch size constant to save both computation and memory use by reducing image size only.

High-resolution finetuning. We perform high resolution finetuning after the main low-resolution training. Intuitively speaking, the model has acquired a high-level understanding of the images and texts through the main training phase. We improve its representation further by providing more detailed visual information through a short high-resolution finetuning process. The images used for high-resolution training are the same as those for the low-res training, except that we remove the downsampling to preserve the rich visual details.

Care is taken to initialize the positional embeddings from low-res pretraining to high-res finetuning. We up-sample the positional embedding weights from the low dimension (*e.g.*4x4 for $h = w = 64$) to the dimension of high-resolution positional embeddings (*e.g.*14x14 for $h = w = 224$) for a given patch size $p = 16$. Compared to up-sampling the low-res positional embeddings without increasing the amount of weights, we found this weight up-sampling beneficial because the

positional embeddings have higher capacity to adapt with more detailed spatial representation. We use trainable positional embedding throughout the paper following existing works [336, 110, 457].

Network architecture. We use the ViT-Large backbone as image encoder by default unless noted otherwise. The ViT-Large is a vision transformer which consists of 24 multi-head self-attention layers with 16 heads and the width dimension of 1024. The patch size is fixed at 16 following common practice. Although we focus on ViT architecture in this study, RECLIP involves only changing the input size, and can potentially support other network architectures as well [413, 110, 162, 274, 400]. Our text encoder follows the same transformer design as previous works [336, 457]. The text encoder consists of 12 multi-head self-attention layers with 12 heads and the width dimension of 1024.

Implementation details. We use a starting learning rate of 0.001, and train for 250k and 550k steps with linear LR decay using an Adafactor optimizer. We set weight decay to 0.01 and batch size to 16384. The batch size is chosen to be a multiple of 1024 and the model feature dimension (e.g. 4096) a multiple of 128, so that TPU padding would not occur on the sequence dimension. A short LR warmup of 2500 steps is used. Our high-resolution finetuning schedule starts with a learning rate of 10^{-4} with 5000 steps LR warmup, and decays linearly over a total schedule of 20k or 50k iterations. We use an image size of 224 or 448 for finetuning. We use the English subset of the WebLI dataset [69] for training. Our training is run on TPU-v3 infrastructure. Compared to general-purpose GPU devices, TPUs are specifically designed for large matrix operations commonly used in neural networks. Each TPU v3 device has 16GB high-bandwidth memory per core, which is comparable to that of a V100 and suitable for synchronous large-scale training. For zero-shot image classification, we use the same text prompts as [336].

3.4 Experimental Results

3.4.1 Main Results

Zero-shot image-text retrieval and image classification. Following existing works [336, 247, 457], we evaluate RECLIP on zero-shot image and text retrieval on Flickr30K [327] and MSCOCO [70] test sets, and zero-shot image classification on ImageNet [94], ImageNet-A [168], ImageNet-R [166], ImageNet-V2 [342] and ImageNet-Sketch [419] datasets. we take each image and text to the corresponding encoder to obtain embeddings for all image and text pairs. Then we calculate the the cosine similarity scores for the retrieval, and use the aligned image and text embeddings to perform zero-shot image classification by matching images with label names without fine-tuning.

Table 3.1 presents the results of RECLIP on this benchmark, where the baseline is our own reproduced version of CLIP. Our baseline model trains on the WebLI dataset with the images of 224×224 for 300k and 600k steps. The original CLIP [336] model and trains on their own dataset with the image size of 336×336, which is marked in gray. RECLIP uses small images for the main training phase and finetune the model with the images of 224×224 for 20k or 50k steps.

For long-schedule training of 600k steps, RECLIP-64 significantly reduces compute use by $\sim$ **6** times from 52.8K to **9.2K** in cores$\times$ hours, which saves $\sim$ **80**% compute resource, and it outperforms the baseline model by +3.9 on Flickr and MSCOCO retrieval. RECLIP-64-F20K, which finetunes the model for only 20k steps with high-resolution images, further reduces the computation use by $\sim$ **8**$\times$ to **6.5K** and improves retrieval performance by +1.8. On zero-shot image classification, RECLIP-64 achieves 75.4 and RECLIP-64-F20K achieves 74.9 of the top-1 accuracy, which is very competitive with the baseline method. RECLIP-64 reduces the token length for the image encoding from 196 to **16** during the main training phase, which is a key factor for resource

savings. Overall, RECLIP-64 shows attractive trade-offs between the resource use and zero-shot retrieval and image classification performance.

Table 3.1: Zero-shot image-text retrieval, image classification results. CLIP*: The original CLIP model [336] is marked in gray. The resource use is converted to TPU-v3 core-hours per [247]. CLIP, our repro.: our reproduced CLIP. RECLIP-X: RECLIP trained with image size X where $X = 64, 80, 112$. RECLIP-64-F20K: RECLIP-64 finetuned for a shorter schedule of 20k steps. Best results are **bolded**.

Method	Training steps	Cores ×hours	Zero-shot Retrieval				Zero-shot INet Classification				
			Flickr30K (1K test set)		MSCOCO (5K test set)						
			I2T R@1	T2I R@1	I2T R@1	T2I R@1	INet	INet-A	INet-R	INet-V2	INet-Sketch
CLIP* [336]	-	120.0K	88.0	68.7	58.4	37.8	76.2	77.2	88.9	70.1	60.2
CLIP, our repro.	300k	26.4K	89.3	75.4	61.3	45.1	**74.5**	54.4	**88.9**	**67.7**	**64.5**
RECLIP-112	300k	13.1K	90.0	76.6	**63.1**	45.0	74.2	55.4	87.8	67.2	63.2
RECLIP-80	300k	7.5K	**91.0**	**77.1**	62.8	**45.7**	74.3	**56.7**	87.8	67.2	62.9
RECLIP-64	300k	6.6K	89.4	77.0	62.2	45.2	73.3	53.7	86.3	66.2	61.6
RECLIP-64-F20K	270k	**3.9K**	88.5	76.1	60.8	44.5	72.6	51.7	85.3	65.3	60.6
CLIP, our repro.	600k	52.8K	89.3	76.9	63.3	46.8	**76.4**	60.2	**90.9**	**70.1**	**66.4**
RECLIP-112	600k	23.4K	90.6	77.6	63.6	46.5	75.8	58.8	89.3	69.1	65.2
RECLIP-80	600k	11.2K	**91.3**	**78.2**	**64.6**	**47.2**	75.8	60.3	89.0	69.2	64.6
RECLIP-64	600k	9.2K	91.0	78.1	64.2	46.9	75.4	**60.9**	88.8	68.9	64.5
RECLIP-64-F20K	570k	**6.5K**	91.0	77.1	63.6	46.2	74.9	58.6	88.2	68.4	63.5

We also train RECLIP with the image size of 80×80. Comparing to the baseline method which consumes 52.8K in cores×hours, our RECLIP-80 remarkably reduces resource usage by $\sim \mathbf{5}$ times to **11.2K**. RECLIP-80 improves retrieval results by +5.0 on Flickr30K and MSCOCO test sets, and achieves highly competitive zero-shot image classification performance of 75.8. Specifically, taking INet-A as an example, RECLIP-80 outperforms the baseline method for both 300k and 600k training steps. For short training schedule with 300 steps, RECLIP-80 requires only **7.5K** in cores×hours which is $\sim \mathbf{4}\times$ less than the baseline model.

Table 3.2: Comparisons of GFLOPs between RECLIP and the baseline model during the RECLIP training.

Models	GFLOPs
CLIP, our repro.	71.4
RECLIP-112	24.8
RECLIP-80	10.1
RECLIP-64	7.3

GFLOPS. We compare GFLOPs of RECLIP with the baseline method in Table 3.2. The baseline method, CLIP, our repro., requires 71.4 GFLOPs. Our RECLIP-80 reduces GFLOPs by $\sim \mathbf{7}\times$ to **10.1** and **RECLIP-64** further reduces GFLOPs by $\sim \mathbf{10}\times$ by using even smaller images.

3.4.2 System-level Comparison

We present system-level comparison between RECLIP and a series of existing methods on Flickr30K and MSCOCO image-text retrieval benchmarks, and ImageNet classification accuracy in Table 3.3. We train RECLIP for 600k steps and then finetune for 50k steps with the image size of 448×448. For RECLIP-64-F20K, we finetune for 20k steps.

Table 3.3: Comparisons of zero-shot image-text retrieval and ImageNet classification top-1 accuracy on Flickr30K, MSCOCO and ImageNet. Models that use the fully-supervised dataset [385] and much larger are marked in gray. †: We refer to [247] to convert GPU cost to TPU usage in CLIP [336], FILIP [454]. Cores×hours results are reported on TPU-v3 infrastructure. Best results are **bolded**.

	Image	Cores		Flickr30K (1K test set)				MSCOCO (5K test set)			
	Encoder	×	ImageNet	image-to-text		text-to-image		image-to-text		text-to-image	
Method	Size	Hours	Top-1	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
PaLI [69]	3.9B	598.7K	85.4	-	-	-	-	-	-	-	-
BASIC [319]	2.4B	288.1K	85.7	-	-	-	-	-	-	-	-
CoCa [457]	1B	962.1K	**86.3**	**92.5**	**99.5**	**80.4**	**95.7**	**66.3**	**86.2**	**51.2**	**74.2**
CLIP [336]	302M	120.0K†	76.2	88.0	98.7	68.7	90.6	58.4	81.5	37.8	62.4
ALIGN [188]	408M	355.0K	76.4	88.6	98.7	75.7	93.8	58.6	83.0	45.6	69.8
FILIP [454]	302M	180.0K†	**78.3**	89.8	**99.2**	75.0	93.4	61.3	84.3	45.9	70.6
FLIP [247]	303M	81.9K	75.8	91.7	-	78.2	-	63.8	-	47.3	-
RECLIP-80 (ours)	303M	28.7K	76.3	91.4	99.1	**79.2**	94.7	**64.9**	**85.2**	**48.2**	**72.6**
RECLIP-64-F20K (ours)	303M	**16.4K**	75.3	**92.5**	99.1	78.7	**94.9**	64.5	**85.2**	47.3	71.9

From Table 3.3, we observe clear resource savings and highly competitive performance achieved with our simple and efficient training recipes. RECLIP with small images saves $3 \sim 59\times$ compute resource in cores×hours. When comparing to the models with the similar scale of the image encoder [336, 188, 454, 247], RECLIP reduces resource use by $\mathbf{5} \sim \mathbf{22}$ times with competitive zero-shot retrieval and image classification performance. In the comparisons to FLIP [247], RECLIP-64-F20K uses $\sim \mathbf{5}\times$ less resource in cores×hours and outperforms it by +2.0 on Flickr30k

and MSCOCO retrieval. Surprisingly, when compared to the CoCa, RECLIP-64-F20K significantly saves $\sim \mathbf{98}\%$ resource use and achieves the best image to text retrieval on Flickr30K test set, giving 92.5 of R@1. RECLIP-64-F20K gives 75.3, which is very competitive on zero-shot ImageNet classification among purely language supervised approaches. We believe this resource savings mostly come from the use of very short image sequence length *i.e.*, 16, which is very different from existing recipes [336, 247, 457, 468].

We also observe that RECLIP-80 uses $3 \sim 34\times$ less compute resource. When comparing to the CoCa [457], RECLIP-80 saves $\sim \mathbf{97}\%$ resource use and achieves highly competitive retrieval performance. The resource savings of RECLIP-80 can also be attributed to the largely-reduced sequence length, *i.e.*, **25** for the image encoding. RECLIP-80 achieves highly competitive ImageNet top1 accuracy of 76.3, which outperforms CLIP and is on-par with ALIGN. Overall, RECLIP provides very affordable recipes for large-scale language and image pretraining.

We note that some leading methods [69, 319, 457] marked in gray demonstrate substantially better zero-shot classification because of larger image encoder capacity and the use of JFT [385] dataset. JFT is a human-annotated classification dataset which is cleaner than most web crawled image-text datasets [336, 365, 188] and most advantageous for zero-shot classification, so we list the JFT-trained entries there for reference only.

3.4.3 Open Vocabulary Detection

We conduct evaluation on the LVIS dataset [153] by using RECLIP for open vocabulary detection. We take a recent SOTA approach RO-ViT [207] as the baseline and apply RECLIP-80 to pre-train the model (RECLIP-RO-ViT). We train only on the LVIS base categories (frequent & common) and test on both the base and novel (rare) categories following the protocol of ViLD [148]. The

results are in the Table 3.4. RECLIP-RO-ViT achieves 32.0 Mask APr (AP on rare categories) [153], matching the state of the art performance of RO-ViT (32.1). This is surprisingly encouraging because detection task typically requires much higher resolution e.g. 1024 than classification task to recognize the small objects, which can be especially challenging for RECLIP due to the low-res information loss. In addition, RECLIP-RO-ViT outperforms RO-ViT by 0.7 on all-category AP, showing that its representation is also suitable for standard detection on the base categories. These detection results suggest that RECLIP representation is versatile and suitable for a broader range of object and pixel-level tasks.

Table 3.4: LVIS open-vocabulary object detection. RECLIP maintains the same open-vocabulary detection (AP_r) and standard detection (AP) as the state of the art RO-ViT despite using much less training resources.

ViT based method	Pretrained model	Detector backbone	AP_r	AP
RO-ViT [207]	ViT-L/16	ViT-L/16	**32.1**	34.0
RECLIP-RO-ViT (Ours)	ViT-L/16	ViT-L/16	32.0	**34.7**

3.4.4 Ablations

In this section, we ablate the design of RECLIP training and evaluate on the zero-shot retrieval and classification accuracy.

The importance of high-resolution finetuning. Table 3.5 shows the importance of finetuning RECLIP with high-resolution data after the main training phase. We compare the retrieval and classification accuracy by using the model trained with and without high-resolution finetuning on an image size of 224 for 50k steps. We observe that high-resolution finetuning significantly improves the performance for zero-shot retrieval and classification. In particular, training RECLIP by using the smallest images, *e.g.*64×64, high-resolution finetuning offers the most notable benefits. This

is also aligned with the results in Table 3.3 where RECLIP models trained with small images, *e.g.*64×64 or 80×80, and finetuned with 448×448 for a short cycle can achieve comparable performance with SOTA models.

Table 3.5: The importance of RECLIP high-resolution finetuning. We found that high-resolution finetuning significantly improves zero-shot transfer performance. RECLIP-X: RECLIP trained with image size X. Best results are **bolded.**

	Total	Before high-resolution finetuning					After high-resolution finetuning				
	Training	INet	Flickr30K		MSCOCO		INet	Flickr30K		MSCOCO	
	Steps	Top-1	I2T	T2I	I2T	T2I	Top-1	I2T	T2I	I2T	T2I
RECLIP-112	300k	**69.0**	**83.2**	**67.9**	**58.6**	**40.0**	74.2 (+5.2)	90.0 (+6.8)	76.6 (+8.7)	**63.1 (+4.7)**	45.0 (+5.0)
RECLIP-80	300k	66.3	80.8	65.4	54.6	37.4	**74.3 (+8.0)**	**91.0 (+10.2)**	**77.1 (+11.7)**	62.8 (+8.2)	**45.7 (+8.3)**
RECLIP-64	300k	62.8	79.6	63.6	51.4	34.5	73.3 (+10.5)	89.4 (+9.8)	77.0 (+6.4)	62.2 (+10.8)	45.2 (+10.7)
RECLIP-112	600k	**70.7**	**87.4**	**71.9**	**59.0**	**40.9**	**75.8 (+5.1)**	90.6 (+3.2)	77.6 (+5.7)	63.6 (+4.6)	46.5 (+5.5)
RECLIP-80	600k	67.7	82.8	68.1	55.8	39.0	**75.8 (+8.1)**	**91.3 (+8.3)**	**78.2 (+10.1)**	**64.6 (+ 8.8)**	**47.2 (+8.2)**
RECLIP-64	600k	65.5	80.9	66.1	54.3	37.1	75.4 (+9.9)	91.0 (+10.1)	78.1 (+12.0)	64.2 (+10.1)	46.9 (+9.8)

Text length for RECLIP main training. Table 3.6 studies the text length for the RECLIP training. We use the text length of 64 and 16 to train our RECLIP with an image size of 80. Somewhat surprisingly, we observe that using a short text length, *i.e.*16, during the main training phase clearly reduces the resource use and achieve competitive zero-shot retrieval and image classification performance. This training efficiency gains is possible because we use much shorter image sequence lengths than existing recipes [336, 457].

Table 3.6: The effect of the text length in RECLIP main training. We found that using a short image sequence can further save compute resource and achieve promising zero-shot transfer performance. Default RECLIP settings are in dark gray. Best results are **bolded**.

Text	Cores	Flickr30K		MSCOCO		INet
Length	× hours	I2T	T2I	I2T	T2I	Top-1
64	15.5K	91.2	78.0	64.3	46.7	75.6
16	**11.2K**	**91.3**	**78.2**	**64.6**	**47.2**	**75.8**

Small batch size for RECLIP main training phase. Our RECLIP is designed with principles of using constant batch size but varying image resolutions during the main training phase. Table 3.7

Table 3.7: The importance of RECLIP main training with constant batch size. We found that using the same batch size (16k) for RECLIP main training and finetuning achieves better zero-shot transfer performance. Default RECLIP settings are in dark gray. Best results are **bolded**.

Batch	Cores×	Flickr30K		MSCOCO		INet
Size	Hours	I2T	T2I	I2T	T2I	Top-1
4k	**4.2K**	81.9	68.8	51.2	38.6	64.4
16k	13.1K	**90.0**	**76.6**	**63.1**	**45.0**	**74.2**

ablates effects of the batch size during the main training phase on zero-shot retrieval and image classification accuracy. We first train the model for 250k steps by using the batch size of 4k or 16k and the image size 112; then we finetune it for 50k steps by using the batch size of 16k and the image size of 224. From Table 3.7 shows that using smaller batch size (4k) saves compute resource by 69%, but the zero-shot retrieval and classification performance drops significantly even with the same high-resolution finetuning phase. Therefore, we conclude that using the same large batch size is important for language image pretraining to ensure competitive zero-shot transfer performance.

Increasing the batch size with small images for RECLIP. In Table 3.8, we ablate RECLIP by varying both the batch size and image size during the main training phase. The multi-grid training paradigm is as below: (1) we equally divide training process into 3 stages with the same steps in each; (2) we train the model for 25k, 50k and 100k steps by using the batch size of 64k, 32k and 16k, and the image size of 112, 160 and 224 in each stage. The idea is to increase the batch size while using low resolution data, and decrease the batch size with high-resolution data. The multi-grid free baseline is trained for 300k steps by using a constant batch size 16k and image size 112, and finetuned with image size 224. We observe that RECLIP without "MG" is not only simpler, but saves computational resource by 30%. In addition, RECLIP achieves better zero-shot retrieval retrieval performance on Flickr30K and MSCOCO and very similar ImageNet performance.

Table 3.8: The effect of multigrid training strategy, where we increase the image size and decrease the batch size simultaneously. We found RECLIP is simple and effective. Default RECLIP settings are in dark gray.

MG	Cores× Hours	Flickr30K		MSCOCO		INet
		I2T	T2I	I2T	T2I	Top-1
✓	18.4K	89.2	75.5	62.3	**45.3**	**74.5**
✗	**13.1K**	**90.0**	**76.6**	**63.1**	45.0	74.2

Multi-stages RECLIP high-resolution finetuning. In Table 3.9, we further study RECLIP with 1 and 2 high-resolution finetuning stages given a model trained with low-resolution data. We study the following two variants. $(112 \rightarrow 224 \rightarrow 448)$: we train the model for 300k steps with the image size of 112, finetune it for 40k steps with the image size of 224, and finetune it for another 40k steps with the image size of 448. $(112 \rightarrow 448)$: we train the model for 300k steps with the image size of 112 and finetune it for 50k steps with the image size of 448. We set 50k steps to keep the computation cost comparable with the first one. We observe that $(112 \rightarrow 448)$ gives very competitive zero-shot retrieval and image classification accuracy. Thus, we use only one high-resolution finetuning stage.

Table 3.9: RECLIP with one-stage or multi-stages high-resolution finetuning. We found that one high-resolution finetuning stage is simple and sufficient. Default RECLIP settings are in dark gray. The best results are **bolded**.

Stages	Core× Hours	Flickr30K		MSCOCO		INet
		I2T	T2I	I2T	T2I	Top-1
$112 \rightarrow 224 \rightarrow 448$	31.1K	91.0	77.7	64.1	**47.4**	**76.2**
$112 \rightarrow 448$	**30.8K**	**90.7**	**78.0**	**64.3**	47.0	76.1

Comparisons of image resizing and token masking Table 3.10, we present a comparison between token masking [247] and image resizing training strategy with matching computational budget. The benchmark is zero-shot ImageNet classification. All factors other than masking vs resizing are controlled to be the same. For example, we use the same batch size, data, training recipe, and the same number of iterations for low-resolution (vs masked) pretraining and high-resolution

(vs unmasked) finetuning. To match the compute usage betweeen resizing and masking, we set the masking ratios such that the sequence lengths are the same. For example, Mask-112 masks 75% tokens to match the sequence length of RECLIP-112 (assuming the baseline using full image size 224x224). Table 3.10 shows that image resizing has a clear advantage over token masking. RECLIP-112 starts with a gap of +2.9 with Mask-112. As the token masking ratio goes above 75% (Mask-112), we observe an increasing gap between resizing and masking (+5.4% for RECLIP-64), showing the clear advantage of resizing in very low-compute settings.

Table 3.10: Comparison of resizing vs token masking on zero-shot ImageNet classification. RECLIP-X: RECLIP with image size X. Mask-X: token masking with the same compute budget as the corresponding RECLIP-X. Resizing consistently outperforms masking, and the gap increases with decreasing compute budget. Best results are **bolded**.

X (Image Size)	Mask-X	RECLIP-X
112	72.9	**75.8 (+2.9)**
80	71.3	**75.8 (+3.5)**
64	69.5	**74.9 (+5.4)**

3.4.5 Visualization

Visualization of small images. In Fig. 3.3, we visualize images at various resolutions paired with their corresponding texts. We observe that small images generally preserve high-level structures of the original images, and contain sufficient visual information for language supervisions. For example, the martial arts, office meeting, concert, and gymnastics scenes are clearly recognizable down to 64×64 resolution. This supports the key insight of our RECLIP training design that leverages small images for the main training phase to save computation.

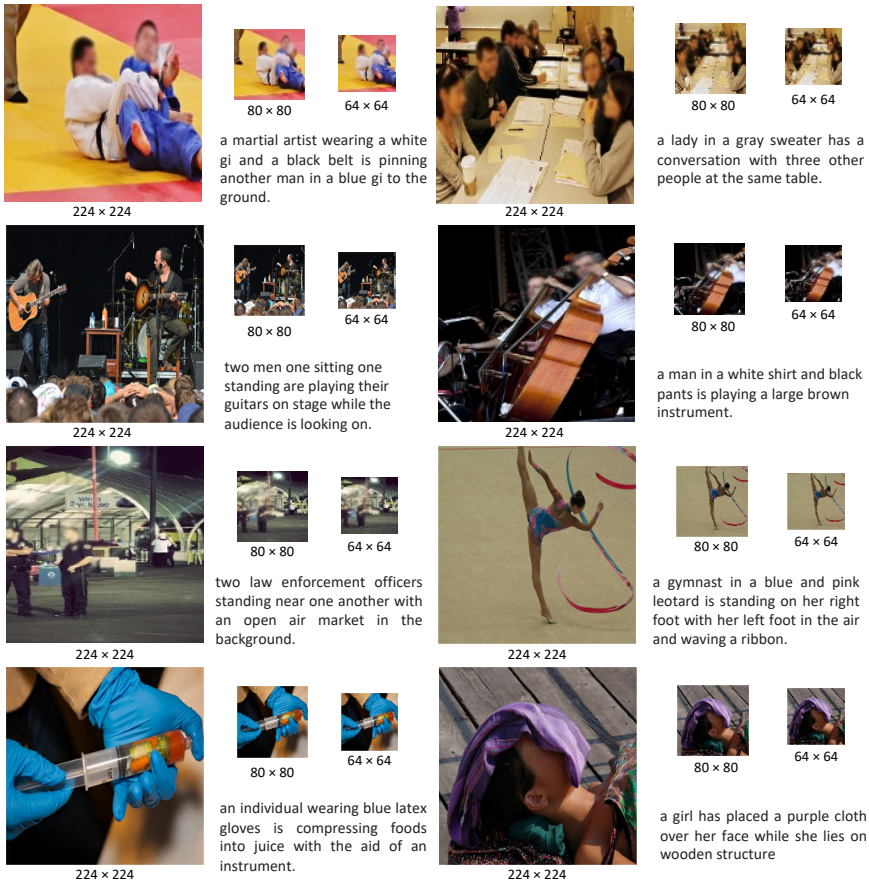

Figure 3.3: Visualization of image-text pairs and images are in various resolutions. Images are scaled with the same factor of 0.01 for both height and width. Small images contain sufficient visual information for contrastive training.

Visualization of image and text retrieval. We present image and text retrieval results of RECLIP in Fig. 3.4. Despite highly resource efficient training, RECLIP still produces accurate results on both image-to-text and text-to-image retrieval. For example, the concepts of football players, race cars, circular sculpture, police officer, musicians, and bulldozer are all correctly matched between image and texts.

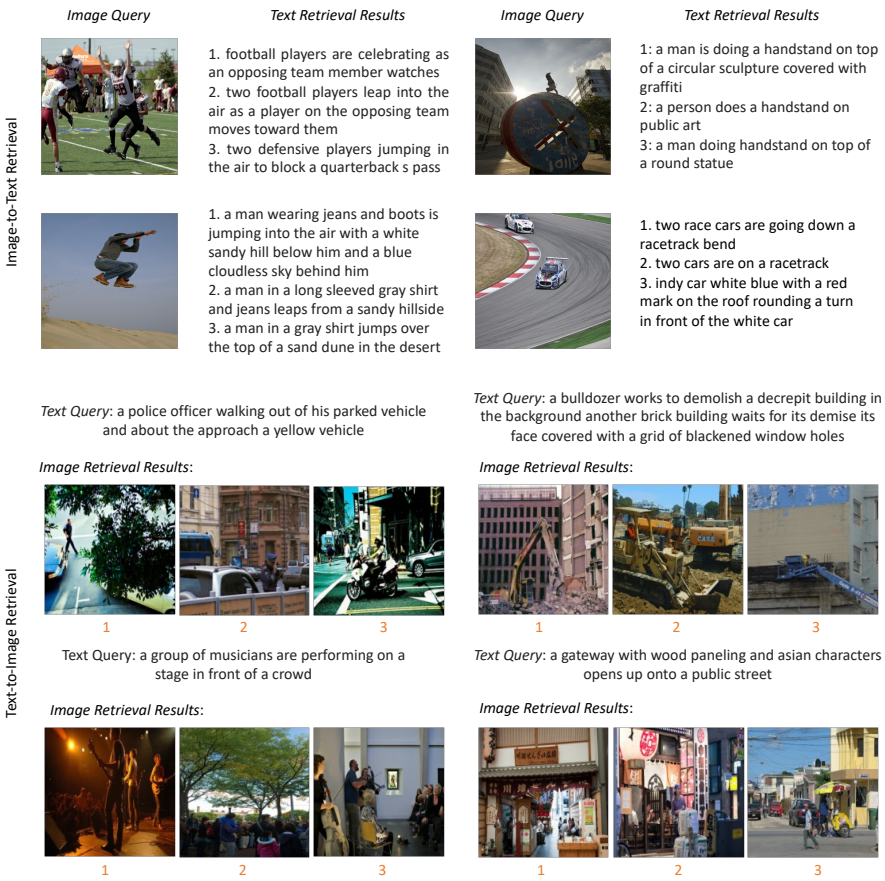

Figure 3.4: Visualization of image and text retrieval results. Despite training with orders of magnitude less resource, RECLIP correctly match many visual concepts with texts.

Broader Impact Statement

Language image pretraining plays an important role in many applications, *e.g.*image and text retrieval, text-to-image generations, open-vocabulary detection, etc. This work presents a language image pretraining method, RECLIP, on large-scale web datasets and the proposed model has been evaluated on a series of zero-shot downstream tasks. The large image-text corpus may contain biased or harmful content which could be learnt by the model. Our model is for research use only and these models should not be used in applications that involve detecting features related to humans (e.g. facial recognition). The good news is RECLIP significantly reduces the resource use,

thereby reducing the carbon footprint and is very environment-friendly for the community to build upon in the long run.

Chapter 4

Face Synthesis With a Focus on Facial Attributes Translation Using Attention Mechanisms

4.1 Introduction

Synthesis of facial attributes and their characterization are important for a wide range of computer vision, forensics, security, biometrics and entertainment applications. For instance, synthesizing face images can be used as a data augmentation technique in order to boost large-scale training for deep neural networks. Face synthesis with facial attributes translation can be applied in attribute-based recognition and identification, especially in surveillance and security. Besides, face synthesis can be useful in digital-art applications, e.g., paintings-related tools, like apps/softwares where users virtually customize faces with various attributes. Further, it is worth exploring how to

extend face synthesis for video generation, *e.g.*, one can generate face videos where facial attributes are changing.

In applications of forensics, security and biometrics, face synthesis with various facial attributes can be applied for disguised and concealed identity recognition by using either synthesized face images or videos. Moreover, adversarial attacks have become hot topics in recent years and synthesizing face images with various facial attributes can be effectively utilized for dealing with adversarial attacks in deep learning models for improving their robustness of deep models. With these broad applications, we argue that translating facial attributes in face images synthesis is an under-explored problem, specifically from the perspective of visual interpretations, because it generally happens as a "black-box" process, without interpretations. However, security critical applications demand a clear understanding of the reasoning behind algorithmic processes. Consequently, there has been substantial recent interest in understanding and interpreting *how* facial attributes are translated in generative models.

Starting from classification tasks, inspired by Zeiler and Fergus [464], much effort has been dedicated in visualizing and understanding feature activations in convolutional neural networks (CNNs) by generating attention maps that highlight regions which are considered to be important for the network goals. As shown in the top stream in Fig. 4.1, given a trained CNN model, attention maps show how CNN responds to input images by indicating where an object is located in an image, *e.g.*, an elephant is classified correctly and highlighted visually in the attention map. CAM [489], Grad-CAM [368] and Grad-CAM++ [57] were proposed to generate this kind of visualization by means of attention maps to help us interpret why this image is classified to the *elephant* class in a more discriminative way.

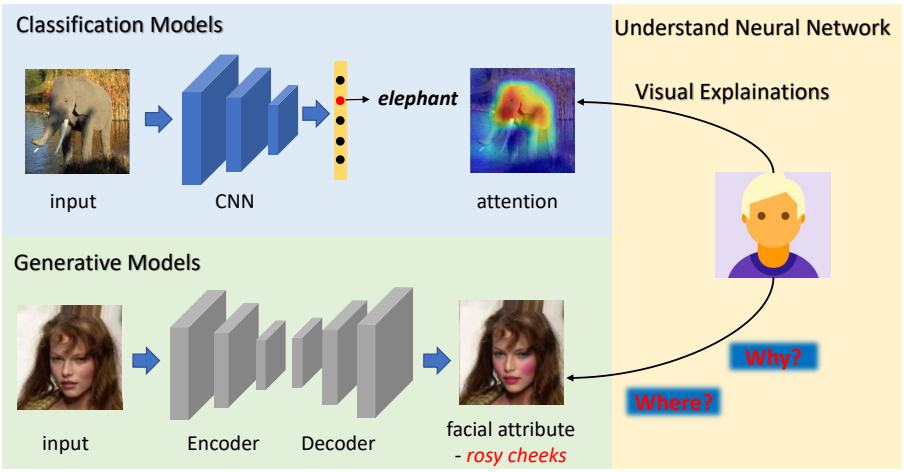

Figure 4.1: Exploring interpretability of traditional CNN models (*top*) and generative models (*bottom*).

Following this line of research, various researchers exploited the model explainability and explored the utilization of valuable information contained in attention maps in order to improve the performance of CNNs. Li *et al.* [237] used attention maps as weak supervised visual guidance for training CNNs and observed improvements in model generalizability in image classification and segmentation tasks. Dhar *et al.* [100] proposed an approach with attention distillation loss for incremental learning for classification tasks. Wenqian *et al.* [269] proposed a technique to visually interpret Variational Autoencoders (VAEs) and utlized attention maps for anomaly localization and disengled representation learning. However, in spite of these significant advances, enabling explainability of GANs by using visual attentions is an area that has not been fully explored yet. For example, how GANs respond to input under different conditions still remains unresolved, which results in GANs being used mainly as a black-box tool.

With the introduction of GANs and their many variants, image generation has made a gigantic forward leap in recent years, especially in terms of photo-realism. Still, often it is necessary

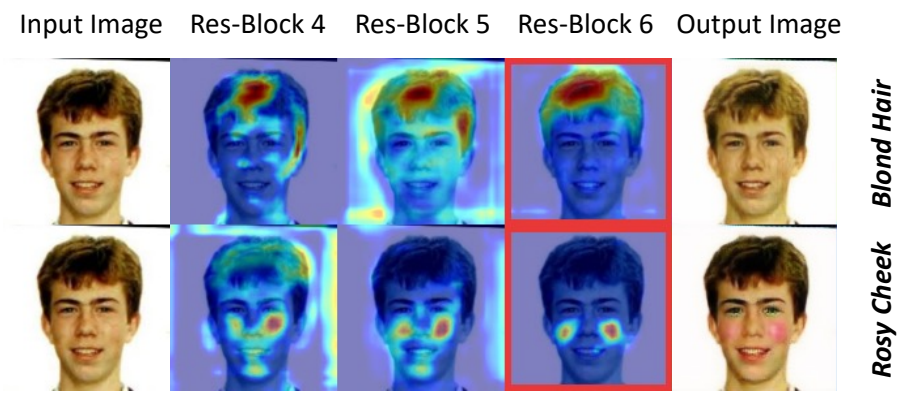

Figure 4.2: Visual interpretations obtained from different residual blocks over the conditional GAN for facial attributes translation. The input to the model is a source face image with target facial attributes and the output is the translated face image where target attributes are expected to be applied on. The attention maps highlight the pixels which contribute to the output class.

to condition the generative models in order to have control over their outputs. This is the case of facial attribute translation, where the objective is to have a more useful representation of input face images that can be later used for various down-streaming tasks. For example, as shown in the bottom stream of Fig. 4.1, generative models that are able to produce highly detailed images with expected facial attributes and their outputs are almost indistinguishable from real face images. Understanding how these generative models perform such kind of tasks for a set of diverse conditions on the input is crucial for us to interpret them.

In this paper, we posit that exploiting visual interpretations in conditional GANs (cGANs) is a fundamental step in order to improve upon them. For this reason, we first study the generation of visual attention in (cGANs) for facial attribute translation by means of a gradient-based method. Facial attribute translation task requires the model to translate a input face image into different face images with specified different facial attributes using only a generator and a discriminator as a GAN system. Then, a fundamental step is taken to produce visual attentions which highlight the spatial

features where the network is focusing on for a certain facial attribute and also allows to identify which layers in the generator are relatively more devoted to the facial attribute translations. Some examples of attentions are presented in Fig. 4.2.

Next, we focus on the utilization of these attention maps to derive a knowledge distillation module in a teacher-student paradigm and propose a novel framework called Attention Knowledge Distillation Generative Adversarial Network (**AKD-GAN**) for facial attribute translation, thus improving performance of translating target facial attributes on input face images. In other words, the teacher network suggests meaningful visual attention for each attribute, that will guide the training of the student network. Further, using attention knowledge distillation helps us in removing biases that are common in facial-attribute translation datasets (*e.g.*, "gender bias" where the selection of a certain attribute also changes the gender of the input face image).

In addition, another flexible application enabled by our proposed model is the use of so-called "pseudo"-attention maps, that are attention maps generated by the teacher from a set of facial attributes and used as weak supervisions for training a different set of facial attributes in order to help the student network to produce better face images with target attributes. This application is developed as a "pseudo"-attention knowledge distillation module in **AKD-GAN**. Extensive experiments are conducted on publicly available datasets and experimental results on two different settings demonstrate the effectiveness of the proposed model.

4.2 Related Work and Our Contributions

4.2.1 CNNs visual attention explanation

Deep Convolutional Networks have achieved astounding results in most computer vision tasks and interpreting their behaviours by visualizing "where they look" when making a decision has attracted lots of interest in the past years. Following the initial work of Zeiler *et al.* [464] and Mahendran *et al.* [288], Zhou *et al.* [489] provided a method for generating *class activation mappings* (CAM) by using the global average pooling. Grad-CAM [368] and its variant Grad-CAM++ [57] were proposed by using gradients of the output score and intermediate feature maps to obtain the gradient-based class-discriminative attention maps. Compared with the response-based approaches [489, 128, 476] which introduce additional trainable units, they are applicable to a wide range of architectures without requiring any structural change in the network and without retraining the models. Recently, the concept of visual attention was also extended to GANs [471, 478, 255]; However, these papers mainly studied self-attention modules which required a large number of additional training units consisting of a series of convolutional layers with 1×1 kernel size. In particular, MU-GAN [478] introduced an additive attention mechanism to build attention-based U-Net connections and a self-attention mechanism in the convolutional layers. In addition, Kim *et al.* [205] calculated attention in the discriminator of a GAN in order to use it as a mask to preserve the attribute-irrelevant regions. As compared to this work, we take the first step into visualizing and employing attention that is calculated from the GAN generator to directly improve the performance of face images synthesis.

4.2.2 Knowledge distillation in neural networks

Knowledge distillation involves transferring knowledge from a more complex network (teacher) to a simpler and lighter network (student) sharing the same task [171]. The goal is to have the student to reach almost the same results as the teacher. Many techniques have been developed in this area [61, 313, 328], with [463] being the first to use attention transfer to improve the performance of a student classification network. Previous methods were almost entirely used for recognition or classification models, while [3] introduced a method working on unconditional GANs. Recently, Li *et al.* [241] proposed a compression algorithm for cGANs using feature distillation and neural architecture search.

4.2.3 Conditional GANs for facial attribute translation

Generative Adversarial Networks (GANs) [145, 198, 200], in their many variations represent the state-of-the-art for photo-realistic image synthesis nowadays. In particular, when a much finer control over the output is required, conditional GANs (cGANs) [298] allow the generation of images from text [344, 473, 474], class labels [38, 102] for natural images, sketches or rich textures [355, 444, 452, 427, 101, 102]. Besides, Di *et al.* [103] studied to synthesize attribute-preserved visible face images from thermal imagery for cross-modal matching. Furthermore, while initially a paired dataset was required [264], CycleGAN [500] showed that a conditional GAN can be successfully trained in an unpaired way. Another relevant feature that most cGANs lack is the ability of producing images belonging to different classes or domains using a single architecture. Some models [179, 265] achieve this by using adaptive instance normalization layers [177] combined with a class-specific encoder and a content-specific encoder whereas StarGAN [77] and its

variants [164, 374, 443, 263] take as input both an image and the target label learning to flexibly execute the translation using only one underlying representation. Unlike the above frameworks, models like StarGANv2 [78] focus on different tasks and share different principles in network design. StarGANv2 designs the architecture by heavily relying on using a mapping network to learn multiple styles and utilizing Adaptive Instance Normalization (AdaIN) layers [177] that can only perform global editing over the input images. Thus, StarGANv2 performs more of a style transfer (like most of AdaIN-based methods) than attributes translation as we focus in this paper. We argue that the purpose of StarGANv2 is significantly different from ours. To perform style transfers, the model is trained to take an image to be transferred and a reference image as the input, and outputs a new image by transferring features of reference image over the image to be transferred, and the output image could be completely different in identities. Fig. 4.3 presents a comparison of results of image-to-image facial attributes translation and style transfer. Fig. 4.3(a) shows an example of facial attributes translation on a face image, where the identity of the face and the appearance of the face are maintained, but only facial attributes are changed. Fig. 4.3(b) shows an example of style transfer, where the identity and almost the entire image have been changed in the output. Indeed, when transferring styles, no single facial attribute is changed, but the style and appearance of the entire image have changed and only the pose of the source image is maintained. In this work, we focus on image-to-image translation for facial attributes translation. Finally, cGANs can also be combined with meta-learning for greater flexibility and robustness [125].

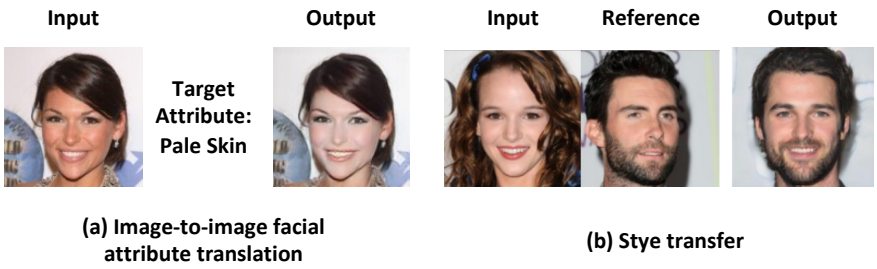

Figure 4.3: Examples of image-to-image facial attribute translation (a) and style transfer (b).

4.2.4 Contributions of this Chapter

- A novel framework called **AKD-GAN** that consists of a teacher and a student network trained with attention knowledge distillation, where visual attentions are designed as full supervisions or weak supervisions to improve the performance of the student network and to remove bias in the datasets.

- An approach that enables to visually interpret conditional generative adversarial networks (cGANs) by using a gradient-based attention mechanism. In addition, the paper shows how these visual attention maps can be used for multiple purposes.

- A demonstration of the proposed method's advantages in improving facial attribute translations with extensive experiments both in distilling attention knowledge among various facial attributes and alleviating observed bias in generated face images.

To bridge the gap of model interpretations of cGANs for synthesizing face images with facial attributes, this work firstly generate visual attention as interpretations for cGANs, and then propose a new framework to utilize visual attention to distill attention knowledge in a teacher-student paradigm to improve the face synthesis performance of the student network. The motivation

comes from our observations that attention maps can highlight spatial regions where the target attributes would be applied (examples in Fig. 4.2), and these attention maps can in turn serve for knowledge-based guidance to further improve model performance. Firstly, we study generating visual attention for facial attributes translation generative adversarial models, *i.e.*, conditional GANs in our framework which has not been explored yet. Secondly, we propose our framework with attention knowledge distillation in a teacher-student paradigm for facial attributes translation and conduct thorough experiments on CelebA [278] and RaFD [229] datasets, establishing improved facial attributes translation performance under extensive experimental settings. Finally, we study how these attention visualizations can help distilling knowledge among *different* facial attributes in our "pseudo"-attention knowledge distillation experiments, providing the flexibility of our proposed attention knowledge distillation module in integrating with generative adversarial networks training.

4.3 Technical Approach

We propose an end-to-end framework, **AKD-GAN** (see Fig. 4.4), to improve model performance of a student network and also of a lightweight student network for facial attributes translation via gradient-based attention maps as guidance. The main idea is to produce visual attention, for facial attribute translation that provides supervisory signals for the proposed attention knowledge distillation process.

We first introduce the backbone network in our **AKD-GAN** and basic training objectives of the generator and the discriminator in the following paragraphs. Then we streamline the attention generation pipeline in Section III-A and the related experimental results are shown in Section IV-D. Next, we describe the proposed attention knowledge distillation loss in Section III-B. The corre-

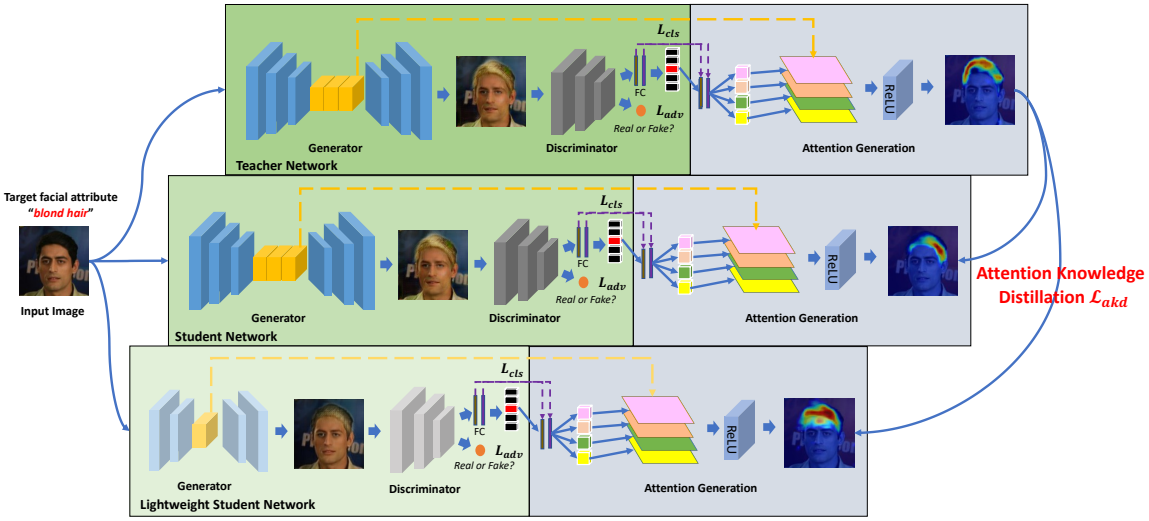

Figure 4.4: Summary of the **AKD-GAN** workflow: the *Teacher* ***T*** network and the *Student* ***S*** network represent the conditional GANs for facial attribute translation. The *Lightweight Student* (***Lite-S***) network is a lighter student network. During training, our proposed attention distillation loss $\mathcal{L}_{akd}$ is calculated using the attention maps obtained from the teacher and the student or the lightweight student network using the method described in 4.3.1.

sponding experiments and discussions are given in Section IV-E, Section IV-G and Section IV-H. In addition, we introduce the proposed "pseudo"-attention knowledge distillation loss in Section III-C and the corresponding experiments and discussions are carried out in Section IV-F.

A conditional generative adversarial network [298] extends GANs by adding a condition as the input to the generator and discriminator. Conditions act as prior information for the GAN to generate data, and such conditions can be in various formats, like latent vectors, images, language priors, etc. The auxiliary classifier GANs [306] further extend a classification stream in the discriminator.

The basic system of our **AKD-GAN** is composed of a teacher ***T*** network and a student ***S*** network which is the one to be optimized and evaluated. Both of them are conditional GANs

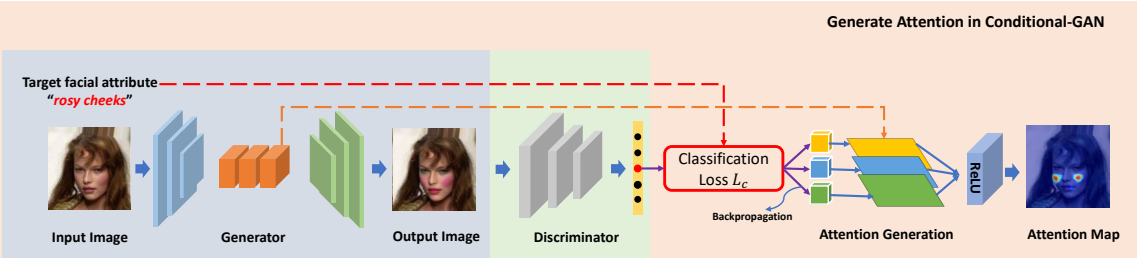

Figure 4.5: Attention generation with conditional GANs for facial attribute translation.

for facial attributes translation and are conditioned by the label of the target facial attribute that we want to translate over the input face image to obtain a new output face image. The generator takes as input a face image and the target facial attribute and returns the translated face image with target facial attributes applied, while the discriminator takes the translated face image as input and returns an adversarial output and a facial attribute classification output.

The facial attribute generator is composed of an encoder, followed by a group of residual blocks and a decoder. The facial attribute translation of the generator can be written as:

$$x_{fake} = Gen_{fa}(x_{real}, c) = Dec(Enc(x_{real}, c)) \tag{4.1}$$

where x_{real} is the input face image, c is the target facial attribute for translation, Gen_{fa} is the facial attribute generator containing the encoder Enc and decoder Dec and x_{fake} is the generated (fake) face image with target facial attributes.

The facial attribute discriminator Dis_{fa} consists of a group of convolutional layers and two output streams: one is the adversarial output telling how realistic the generated face image is by distinguishing it as real or fake, and the other one is auxiliary classification output for calculating the facial attribute classification loss. The overall learning objectives to train the student network of

AKD-GAN for facial attribute translation follows the one of StarGAN [77] and can be written as:

$$\mathcal{L}_{Dis_{fa}} = \mathcal{L}_{adv} + \mathcal{L}_{cls} \tag{4.2}$$

$$\mathcal{L}_{Gen_{fa}} = \mathcal{L}_{adv} + \mathcal{L}_{cls} + \mathcal{L}_{rec} \tag{4.3}$$

for the discriminator and the generator. The generator is trained using adversarial loss $\mathcal{L}_{adv}$, classification loss $\mathcal{L}_{cls}^{Gen_{fa}}$ and reconstruction loss $\mathcal{L}_{rec}$, while the discriminator is trained with adversarial loss $\mathcal{L}_{adv}$ and classification loss $\mathcal{L}_{cls}^{Dis_{fa}}$.

4.3.1 Attention Generation for Facial Attributes Translation

Inspired by the fundamental framework of Grad-CAM [368], we streamlined the generation of attention map on either the student network or the teacher network for facial attribute translation. An attention map corresponding to the input face images can be obtained within each inference step so that it can be employed during training stage. Given an input face image $x \in \mathbf{x_{real}}$ and a set of target facial attributes $\mathbf{c}$, for each class $c \in \mathbf{c}$, from the ground-truth labels of target facial attributes, we compute the gradient of score y^c corresponding to the class c. We backpropagate the gradients directly from the classification output of the discriminator to the convolutional layers of the generator with feature maps $\mathbf{F} \in \mathbb{R}^{n \times h \times w}$, with n, h and w being number of channels, height and width of the feature map, respectively, obtaining facial attributes attention maps $\mathbf{A}^{fa}$ corresponding to y^c. Indeed, this represent a significant difference with respect to an ordinary classification network (the typical setup where Grad-CAM operates) where, in order to get attention, the gradients are backpropagated only to the layer before the classification output. Specifically, we calculate $\mathbf{A}^{fa}$ by using the following equation:

$$\mathbf{A}^{fa} = ReLU\left(\sum_{k=1}^{n} \alpha_k^c \mathbf{F}_k\right) \tag{4.4}$$

where the scalar $\alpha_k^c = \text{GAP}\left(\frac{\partial y^c}{\partial \mathbf{F}_k}\right)$ and $\mathbf{F}_k$ is the k^{th} feature channel ($k = 1, \ldots, n$) of the feature maps $\mathbf{F}$, with $\frac{\partial y^c}{\partial \mathbf{F}^k}$ representing the gradient of the score y^c with respect to the feature maps $\mathbf{F}^k$. The global average pooling (GAP) operation is used to obtain scalar α_k^c as:

$$\alpha_k^c = \frac{1}{S} \sum_{m=1}^{h} \sum_{n=1}^{w} \left(\frac{\partial y^c}{\partial F_k^{mn}} \right) \tag{4.5}$$

where $S = h \times w$ and F_k^{mn} is the pixel value at location (m, n) of the $h \times w$ matrix $\mathbf{F}_k$. The attention map generation process is illustrated in Fig. 4.5.

Note that we took this conditional GAN for facial attributes translation as the main case study in our work, but this pipeline to generate attention maps can be applied to a wider variety of GANs, and simply requires a generator-discriminator structure with auxiliary streams integrated in the discriminator.

Example attention results are shown in Fig. 4.2. It can be observed that visual attention reveal how the conditional generative models perform translations on the input to generate output in a more transparent way. Particularly, these results confirm that we can use these attention maps for knowledge distillation, either in a self-supervised or weakly-supervised manner, to train our **AKG-GAN** for improved facial attributes translations, which will be introduced in following sub-sections.

4.3.2 Attention Knowledge Distillation Loss

We use the notion of attention knowledge distillation as the main part of our learning process for distilling the knowledge encoded in visual attention from the teacher to the student and propose a new learning objective $\mathcal{L}_{akd}$ (attention knowledge distillation loss, denoted as akd loss). Essentially, given the input face images x and target facial attribute c, the attention maps of the facial attribute class are computed from the teacher network and student network using the method

introduced in Sect. 4.3.1, as $\mathbf{A}_{\boldsymbol{T}}^{fa}(x,c)$ and $\mathbf{A}_{\boldsymbol{S}}^{fa}(x,c)$, respectively. We enforce the two attentions to be consistent (*i.e.*, the student attention must imitate the teacher one) and integrate this constraint during the training. To this end, we propose a loss function $\mathcal{L}_{akd}$ which is defined as:

$$\mathcal{L}_{akd} = \mathbb{E}_{x,c}\left[\parallel \mathbf{A}_{\boldsymbol{T}}^{fa}(x,c) - \mathbf{A}_{\boldsymbol{S}}^{fa}(x,c) \parallel\right] \tag{4.6}$$

The proposed loss is differentiable which means that it can be directly used for model training without introducing additional training units.

The **AKD-GAN** workflow is illustrated in Fig. 4.4, with $\mathcal{L}_{akd}$ integrated in different teacher-student knowledge distillation designs: the first one is teacher-student training while the second one is teacher-lightweight-student training. In addition, during each training step of the student network, we trained the student discriminator to distinguish between real and fake face images and to correctly classify the face images into multiple facial attributes. The generator is trained to fool the discriminator by producing better face images belonging to the target facial attributes. Finally, the full training objective for student network is:

$$\mathcal{L}_{Dis_{fa}}^{\boldsymbol{S}} = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls} \tag{4.7}$$

$$\mathcal{L}_{Gen_{fa}}^{\boldsymbol{S}} = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{akd}\mathcal{L}_{akd} \tag{4.8}$$

where $\mathcal{L}_{Dis_{fa}}^{\boldsymbol{S}}$ is the loss function that we use to optimize the discriminator of the student network and $\mathcal{L}_{Gen_{fa}}^{\boldsymbol{S}}$ is the loss function used to optimize the generator of the student network. During the training, only parameters of the student network are optimized and the discriminator and the generator are optimized jointly. Following StarGAN [77], we use $\lambda_{cls} = 1$, $\lambda_{rec} = 10$ in all our experiments. For λ_{akd}, we use different values and empirically set it as 10 which gives us the best results in the preliminary experiments for face synthesis with 5 facial attributes.

The intuition of $\mathcal{L}_{akd}$ is that knowledge involved in visual attentions can be distilled as supervisory signals via a teacher-student paradigm so that training of the student can be boosted. Especially for those facial attributes that need to be translated over a small region of face images, by using $\mathcal{L}_{akd}$, our expectation is that it can help to prevent noise from other face areas so that the model is pushed to focus only on the region where the facial attribute needs to be translated.

4.3.3 Pseudo-Attention Knowledge Distillation Loss as Weak Supervision

In the previous section, we discussed how we have generated attention maps as interpretations and designed the attention distillation loss integrated with the adversarial objectives to train the model for facial attributes translation. In this section, we will discuss how to distill knowledge from a teacher ***T*** network translating input face images with a set of facial attributes to a student ***S*** network that works on a ***new*** set of ***different*** facial attributes. Our intuition is that when the input face images are translated to a different target facial attribute, the regions "looked at" by the model in the input images might be shared through different facial attributes. For example, in order to translate an input face image to the output images with specified facial attributes such as *black hair*, *blond hair* or *brown hair*, the network is expected to pay more attention to and edit the region corresponding to the facial attribute ***"hair"*** in the input image so that it can perform translations in "color". Given these observations, in this section, we start from generating the "pseudo"-attention maps for facial attribute translations and present how to design "pseudo"-attention knowledge distillation loss. Finally, we demonstrate the advantages of using "pseudo"-attentions in designing weak supervisory signals which can be integrated flexibly in training a student network with **AKD-GAN**.

Training AKD-GAN with Pseudo-Attention Knowledge Distillation: The objective is to distill knowledge using "pseudo"-attentions as weak supervisions.

Firstly, given a set of target facial attributes $\mathbf{c}$, we identified a second set of facial attributes $\mathbf{c^{pse}}$ where $c^{pse} \in \mathbf{c^{pse}}$ would share spatial features on the face with a facial attribute $c \in \mathbf{c}$. Next, we trained a teacher $\mathbf{T_{pse}}$ network using the *different* facial attributes $\mathbf{c^{pse}}$. Finally, we defined a "pseudo" attention knowledge distillation module for training **AKD-GAN** using the teacher-student paradigm.

While training the student network using $\mathbf{c}$ as target facial attributes, we generated attention maps and proposed the "pseudo" attention knowledge distillation loss to distill knowledge from the teacher network to the student network. Specifically, given an input face image $x \in \mathbf{x}$ and a target facial attribute $c \in \mathbf{c}$, we generated the attention maps using the method in Section 4.3.1. For the teacher network trained with $\mathbf{c_{pse}}$, the class score $y^{c_{pse}}$ is backpropagated as usual to calculate the facial attribute attention map as $\mathbf{A}_{\boldsymbol{T_{pse}}}^{fa,c^{pse}} = ReLU\left(\sum_{k=1}^{n} \alpha_k^{c^{pse}} \mathbf{F}_k\right)$. For the student network trained to translate to different facial attributes $\mathbf{c}$, the attention for the student network is generated as $\mathbf{A_S}^{fa,c}$. Thus, "pseudo"-attention maps are obtained as $\mathbf{A}_{\boldsymbol{T_{pse}}}^{fa,c^{pse}}$ from the teacher network and $\mathbf{A_S}^{fa,c}$ from the student network, respectively. Since our goal is to train the student network to translate input face images to target facial attributes $\mathbf{c}$, while the supervisions via the knowledge distillation are defined with different facial attributes $\mathbf{c^{pse}}$, we call the attentions obtained in this way "pseudo"-attentions. Finally, the attention knowledge distillation loss $\mathcal{L}_{akd}^{pse}$ employed to train **AKD-GAN** with "pseudo"-attentions is calculated as:

$$\mathcal{L}_{akd}^{pse} = \mathbb{E}_{x,c}[\parallel \mathbf{A}_{\boldsymbol{T_{pse}}}^{fa,c^{pse}}(x, c^{pse}) - \mathbf{A}_{\boldsymbol{S}}^{fa,c}(x, c) \parallel] \tag{4.9}$$

The training objective of discriminator is the same as Equation 4.2 and the objective of generator in student network is:

$$\mathcal{L}_{Gen_{fa}}^{\boldsymbol{S}} = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{akd}^{pse}\mathcal{L}_{akd}^{pse} \tag{4.10}$$

4.4 Experimental Results

4.4.1 Experimental Settings

We present experiments using two different settings to train the **AKD-GAN**: **(a)** we train the student network to translate a set of facial attributes and distill knowledge using the attention maps calculated from a teacher network trained on the same set of attributes, **(b)** we train the student network to translate a set of facial attributes and distill knowledge using the pseudo-attention maps calculated from a teacher network trained on a set of different facial attributes. In each experimental setting, firstly we train the teacher network which shares the same architecture design as StarGAN. Then, we train the student by distilling attention knowledge from the last residual block of the pre-trained teacher network to the last residual block of student networks in all experiments. More in detail, we experimented with student networks having different model complexity: **(a)** a student network (***S***) that shares the same architecture design as the teacher network, **(b)** a lightweight student (***Lite-S***) network which is a lighter and pruned network.

In the lightweight student (***Lite-S***) network, we reduced feature map dimensions and the number of residual blocks, obtaining a much lighter network: the number of feature maps in each layer is one half that of each layer of the teacher generator (*e.g.*, from 64 to 32 feature maps in the first convolutional layer of of (***Lite-S***). Besides, there are only 3 residual blocks instead of 6. Table 4.1 shows the number of parameters of generators in the teacher (***T***), the student network (***S***) and the lightweight student network (***Lite-S***).

In all experiments, only the parameters of student network are optimized during the training and only the student network is utilized during evaluation.

Table 4.1: Number of parameters of different generators.

AKD-GAN Model Design	Number of parameters
Teacher (***T***), Student (***S***)	8.4 Million
Lightweight Student (***Lite-S***)	1.2 Million

4.4.2 Datasets

CelebA: The core experiments were performed using the CelebA dataset [278], which is a large-scale facial attributes dataset with more than 200,000 images and 40 attributes. Facial attributes were also well suited to prove the efficacy of our system, since, in order to correctly translate the input face images to outputs, the network needs to learn spatial information on the human face precisely and visual attentions can acquire this kind of information, especially for small, localized attribute.

RaFD: This dataset [229] includes 8,000 images divided in 8 emotional expressions. Following [77], in all experiments, input images are cropped and resized to 128×128.

4.4.3 Metrics

The classification accuracy of translation is used as the main evaluation metric to evaluate the performance of translating target facial attributes over input face images. To elaborate, we use the training set of each dataset to train a deep attribute classification model (ResNet50 [162]) for the facial attributes to be translated and take the generated face image with target facial attributes as the input to the classifier and calculate classification results. It is noted that the classifier we used for evaluation only saw training samples and never saw testing samples. On the **CelebA** dataset, the classification model achieves an accuracy of 91.3% for all facial attributes on the test set. On the

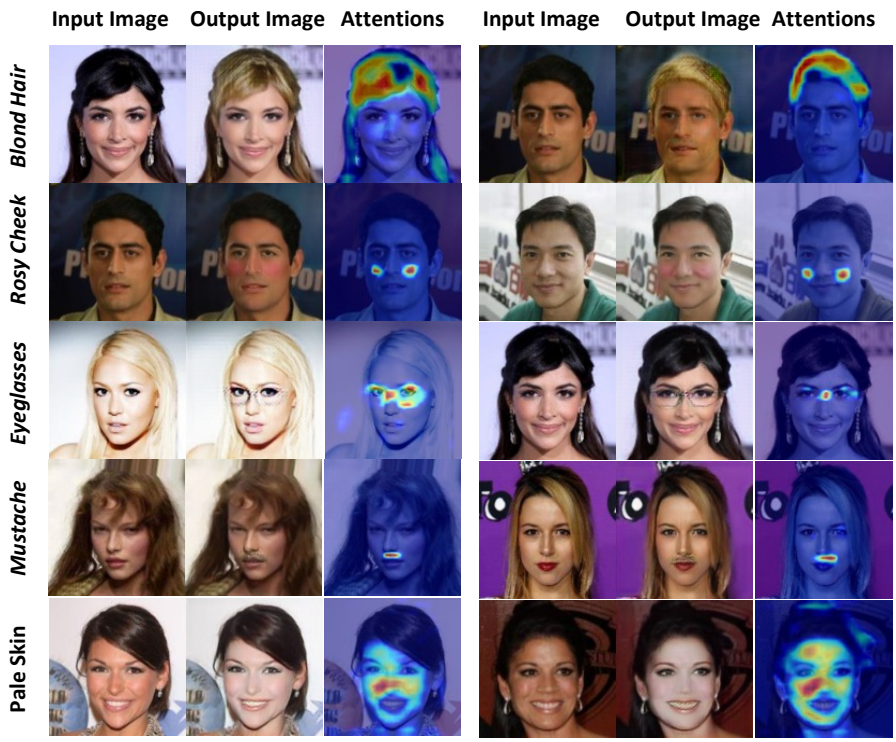

Figure 4.6: Attention maps used by the proposed attention knowledge distillation for facial attributes translation.

RaFD dataset, the classification model achieves an accuracy of 98.9% for all facial expressions. In addition, we resort to use another alternative measure Fréchet Inception Distance [169] to evaluate the image quality.

4.4.4 Visualizing Attentions in Conditional GANs

Examples of visual attention maps for facial attributes translation on the **CelebA** dataset are shown in Fig. 4.6. The generated attention maps are utilized to develop our attention knowledge distillation loss (see Section 4.3.2 and Equation 4.6) and "pseudo"-attention knowledge distillation loss (see Section 4.3.3 and Equation 4.9) for face synthesis with facial attributes translation. As

an example of a comparison, we generate attention maps from style transfer models (see Fig. 1 in the supplementary materials) which target on different tasks from ours and these differences are discussed in Section 7.2. In Fig. 4.6, each triplet of images consists of an input face image, a translated face image and the corresponding attention maps. These attention maps indicate spatial features where the conditional generative models focus on when performing the facial attribute translation. For example, "face" is highlighted precisely when the facial attribute *Pale Skin* is used as the target for translation. In addition, for small attributes, like *Mustache* and *Eyeglasses*, attention maps can localize their spatial areas over the input face images accurately.

It is observed that there are green color artifacts in the generated face images (see Fig. 4.6). They represent a limitation of the state-of-the-art networks in generating realistic texture. We adopted the network designs of StarGAN and we visualized interpretations with attention maps that are obtained directly from StarGAN's generated images. They may be caused by the group of deconvolutional layers in the decoder of the generator. Another reason can lie in the low-resolution data used for this task. Face images are 128×128 and contain rich facial attributes. This poses challenges in synthesizing face images with certain facial attributes translated while maintaining the rest of the face untouched. In the future work, a solution to limit these artifacts would be to replace deconvolutional layers with convolutional layers and interpolation operations. Another empirical solution would be to design the model and training schemes by introducing ideas of the progressive GANs [197].

4.4.5 Attention Knowledge Distillation for Face Synthesis with Facial Attributes Translation

We first conducted experiments by training the proposed **AKD-GAN** with a standard teacher-student paradigm, expecting improved performance of student network on facial attributes translation. Moreover, we trained our **AKD-GAN** by using a light-weight student network where model complexity is reduced significantly, to further evaluate the capability of the attention knowledge distillation module in improving an even smaller model performance. Experiments are conducted on the **CelebA** dataset.

Table 4.2: Quantitative comparisons in classification accuracy between the proposed method (**AKD-GAN**) and state-of-art methods for generated face images over various facial attributes. The bold results represent the best results. Inference in AKD-GAN is performed using only the generator of the student network.

Methods	Interpretations	Blond Hair ↑	Goatee ↑	Eyeglasses ↑	Heavy Makeup ↑	Pale Skin ↑
ACGAN [306]	✗	73.2%	53.2%	95.6%	60.8%	83%
RelGAN [443]	✗	57.31%	65.35%	97.87%	47.54%	52.56%
AttGAN [164]	✗	35.53%	56.90%	98.23%	55.63%	80.22%
STGAN [263]	✗	75.38%	68.22%	95.83%	34.44%	71.78%
StarGAN [77](baseline)	✗	80.94%	64.54%	99.10%	86.27%	79.66%
StarGAN [77](double iterations)	✗	83.68%	71.22%	98.82%	65.57%	83.06%
AKD-GAN (Ours)	✓	**86.02%**	**74.45%**	**99.48%**	**91.47%**	**88.32%**
	Interpretations	Brown Hair ↑	Bushy Eyebrows ↑	Wear Hat ↑	Wear Lipstick ↑	Pointy Nose ↑
ACGAN [306]	✗	76.5%	90.2%	74.4%	61.9%	77.6%
RelGAN [443]	✗	35.20%	62.86%	48.78%	49.29%	20.68%
AttGAN [164]	✗	48.38%	72.37%	20.23%	65.35%	50.99%
STGAN [263]	✗	87.33%	88.25%	35.61%	41.09%	59.61%
StarGAN [77](baseline)	✗	75.10%	95.60%	79.66%	95.92%	82.44%
StarGAN[77](double iterations)	✗	75.22%	87.30%	92.89%	98.00%	88.24%
AKD-GAN (Ours)	✓	**92.71%**	**97.04%**	**92.89%**	**98.00%**	**88.24%**

AKD-GAN with Teacher and Student Networks: In the first set of experiments the objective is to use visual attention to distill knowledge from a teacher $\boldsymbol{T}$ network to a student $\boldsymbol{S}$ network (see top two rows in Fig. 4.4) to improve the performance of student model for facial attribute translation.

We trained the **AKD-GAN** to translate various facial attributes including *blond hair*, *goatee*, *eyeglasses*, *pale skin*, etc. We present both quantitative and qualitative results in the following paragraphs. As introduced in Sec. 4.4.1, our AKD-GAN is built upon StarGAN which, for this reason, is the baseline method. We also tested and compared results with ACGAN [306], AttGAN [164] RelGAN [443] and STGAN [263].

Quantitative Results. Quantitative comparisons of facial attributes classification accuracy are shown in Table 4.2. We used the synthesised face images outputted from the generator in the student network to calculate the classification accuracy of translated images of each facial attribute to evaluate the quality of translations that are made to the input face images. From the Table 4.2, it can be observed that the proposed **AKD-GAN** model trained with attention knowledge distillation loss outperforms all other methods in translating facial attributes in terms of classification accuracy, which proves the effectiveness of our model in translating facial attributes over input faces.

Next, we calculated the FID score using translated face images of each attribute and the results are shown in Table 4.3. Compared with the baseline method, StarGAN, our framework can consistently obtain better quality of face images after translating facial attributes over inputs, which validates that visual attention can help to improve facial attributes translations as well as image generation. Compared with the state-of-the-art methods, our method can achieve competitive performance in synthesizing face images with target facial attributes. In particular, for facial attributes *heavy_makeup*, *wear_hat* and *wear_lipstick*, our method can obtain the best quality of generated face images by large margins.

Table 4.3: Quantitative comparisons in FID score between the proposed **AKD-GAN** and state-of-the-art methods for generated face images over various facial attributes. The bold and underlined results represent the best and the second best results respectively.

Methods	Interpretations	Blond Hair ↓	Goatee ↓	Heavy Makeup ↓	Brown Hair ↓
ACGAN [306]	✗	42.77	71.29	33.62	29.19
RelGAN [443]	✗	**29.04**	**28.34**	40.25	**18.1**
AttGAN [164]	✗	36.13	61.48	<u>29.27</u>	<u>19.13</u>
STGAN [263]	✗	34.5	<u>49.04</u>	36.8	19.53
StarGAN [77] (baseline)	✗	34.87	52.89	62.44	23.51
AKD-GAN (Ours)	✓	<u>30.03</u>	50.87	**21.72**	19.63

Methods	Interpretations	Bushy Eyebrows ↓	Wear Hat ↓	Wear Lipstick ↓	Pointy Nose ↓
ACGAN [306]	✗	33.78	100.47	33.95	27.93
RelGAN [443]	✗	<u>25.52</u>	<u>90.18</u>	34.17	<u>18.53</u>
AttGAN [164]	✗	27	100.15	25.43	20.83
STGAN [263]	✗	**23.35**	118.65	36.06	**16.83**
StarGAN [77] (baseline)	✗	30.92	90.337	<u>23.81</u>	21.53
AKD-GAN (Ours)	✓	28.36	**79.24**	**21.29**	19.48

Given the above observations, it is shown that our method with knowledge distillation loss, which is derived from attention interpretations, can achieve high performance in translating facial attributes over the face images. Meanwhile, our method can generate face images by translating facial attributes with competitive performance.

Qualitative Results. Collections of qualitative results between the proposed framework and the baseline method StarGAN are shown in Fig. 4.7. Each triplet presents the input face image, the generated result from StarGAN [77] and the result from the proposed method, **AKD-GAN**. Indeed, in face images that are generated from the proposed AKD-GAN, target facial attributes are applied much more strongly over the input. In addition, for the facial attributes that require precise spatial localizations for translations, (*e.g. eyeglasses* and *smiling*), **AKD-GAN** gives more convincing results. Furthermore, it is observed that results from the **AKD-GAN** are less noisy and

Figure 4.7: Comparisons of qualitative results between the baseline method (StarGAN [77]) and the proposed **AKD-GAN**.

with fewer artifacts. For example, in the first triplet of the first row of Fig. 4.7, the attribute *blond hair* is applied very convincingly while still maintaining the original face characteristics; in the second triplet of the last row of Fig. 4.7, the attribute *pale skin* is applied well with fewer artifacts.

In addition, Fig. 4.8 presents collections of qualitative comparisons between the proposed method and the STGAN [263]. It can be observed that our method generates face images with target facial attributes applied better than STGAN, in particular, for facial attributes *mustache*, *eyeglasses*, etc.

Empirical Observations on Bias Removal in Facial Attributes Translation: An observation that often occurs in facial attributes translation given a certain dataset, *e.g.*, CelebA, is to find some kind of correlation between attributes (a phenomenon often increased by datasets bias). This means that when translating a certain attribute, other undesired attributes will be translated as

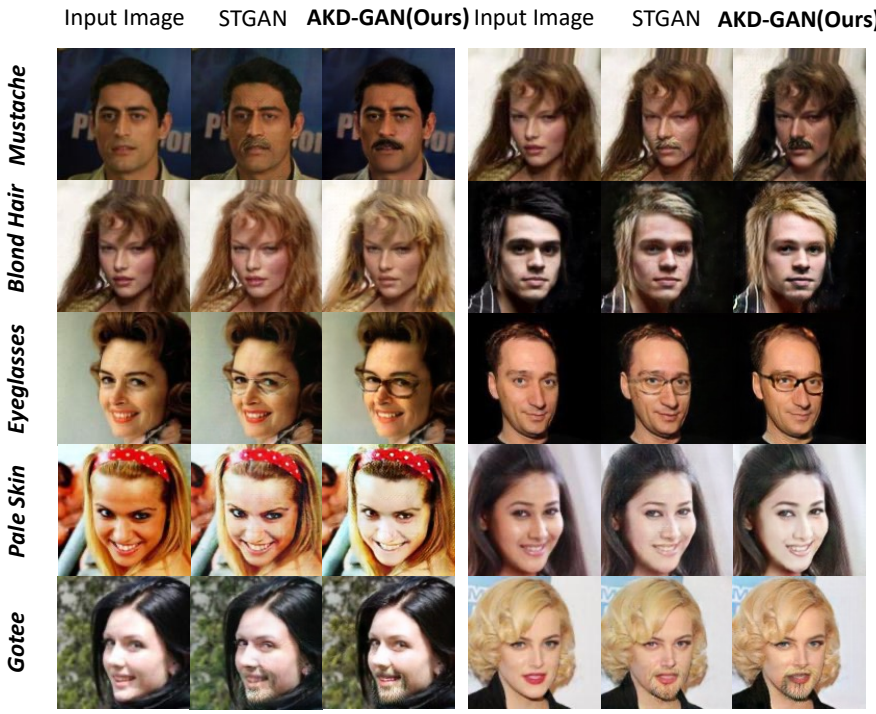

Figure 4.8: Comparisons of qualitative results between the STGAN [263] and the proposed **AKD-GAN**.

well. A clear example is represented by the *Bald* attribute which can only be found in face images of the gender-"male" in the dataset. This means that a facial attribute translation model (like StarGAN) will likely turn each input image into a face image with the gender-"male" when applying the *Bald* facial attribute since it has learned to correlate that attribute with a specific gender.

None of existing works [77, 164, 443, 263] has discussed this phenomenon, while, we argue that it is an important problem and attention should be paid to it. We conduct experiments to evaluate the model performance in mitigating biases on facial attribute translations. Firstly, we use face images in the training set to fine-tune a classifier for face images with attributes *Male* and *Female*. Secondly, we use only face images with the attribute *Female* in **AKD-GAN** and StarGAN to translate input face images to the output with the attribute *Bald*. Finally, to fairly evaluate the

model performance in generating unbiased face images, the aforementioned classifier is used to distinguish if the translated face images have maintained the correct *Female* attribute during the translation. Table 4.4 shows the classification errors of *Female* images being classified as *Male*. It demonstrates that existing state-of-the-art method StarGAN [77] caused an error of over 97%, which means genders of generated face images have largely changed and the facial attribute translation process is completely biased. The classification error can be reduced to around 51% by using the proposed **AKD-GAN**, which proves the capability of the proposed method in eliminating biases for facial attribute translation (by concentrating only on the image parts of interest). Furthermore, a visual comparison is presented in Fig. 4.9 where it is clear how our method does not change the gender of the input face images during the translation. In addition, a dual experiment has been conducted in order to generate males with attribute *Lipstick* which is uncommon in the dataset. The results (presented in last column of Table 4.4) further confirm the efficacy of our method.

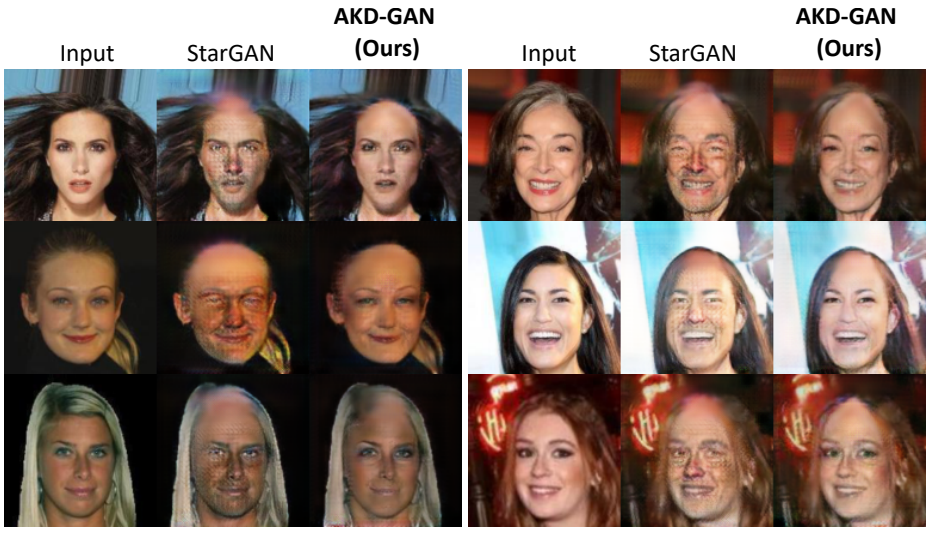

Figure 4.9: Generated face images of the facial attribute *Bald* using StarGAN and our method **AKD-GAN**. AKD-GAN does not change the gender of the input image.

Table 4.4: Percentage of gender **misclassification** for generated face images with target attributes *Bald* and *Lipstick*.

Methods	**Female misclassified as Male (Attribute: Bald)** $\downarrow$	**Male misclassified as Female (Attribute: Lipstick)** $\downarrow$
StarGAN **(Baseline)**	97.72%	88.11%
AKD-GAN (Ours)	**50.97%**	**83.60%**

Nevertheless, improvements of *Bald* and *Wearing Lipstick* are different and we argue that they are related to the significant different distributions of face images with these facial attributes in the dataset. Table 4.5 shows that, (i) over 99% of images with the *Bald* facial attribute are males (ii) over 99% of images with the *Wearing Lipstick* are females (iii) the *Bald* class is very underrepresented in the dataset. In addition, Table 4.5 shows how (i) a large portion of female face images have the attribute *Wearing Lipstick* internally, while, (ii) only a small portion of male images have the attribute *Bald*.

Table 4.5: Distributions of facial attributes *Bald* and *Wearing Lipstick* in the CelebA dataset.

	Bald	Lipstick		Male	Female
% in CelebA	2.2%	47.24%	% in CelebA	41.6%	58.4%
Male/ Female	99.6%/0.4%	99.4%/0.6%	Bald/ Lipstick	99.6%/0.4%	99.4%/0.6%

As a consequence of all these observations, for the *Bald* attribute, even if for the generator it is difficult to apply it over female images, its entanglement with the attribute *Male* is less strong due to a higher number of male images that are not bald. For this reason, localizing the facial attribute for translation using the proposed attention knowledge distillation greatly boosts the performance of generating face images of the *Bald* facial attribute effectively and empirical observations showed the bias mitigation during face image synthesis. Since the percentage of images with the *Wearing Lipstick* facial attribute is much higher in the dataset, and almost every female

image has this attribute, the attribute is much more entangled with the gender and, therefore, the bias is more difficult to mitigate.

We have observed that biases exist as part of the dataset itself. We proposed two potential ways to further tackle this issue in the future. First, one possible way would be to bring external data in order to help mitigate biases in training the model. The external data could be from an extra dataset or data generated by data augmentation tools. Second, another possible way would be to design an online training strategy which enables the model to reject samples with strong biases, especially at the beginning of the training, and accept samples with biases progressively after the model has been trained stably. Our expectation is that the model can be trained without introducing biased data at the beginning, and then trained with introducing biased data to increase the diversity of the data and improve model generalizability.

AKD-GAN with the Teacher Network and Lightweight Student Network: In this set of experiments the objective is to use visual attentions to distill attention knowledge between a teacher ***T*** and a smaller, lightweight student (***Lite-S***) network. Our expectation is that reduced complexity of model can be remedied by attention knowledge distillation. The models were trained to translate different facial attributes: *goatee*, *rosy cheeks*, *eyeglasses*, *mustache*, *blond hair*, etc. We present both quantitative and qualitative results in following paragraphs.

Qualitative Results. Qualitative results are presented in Fig. 4.10. Thanks to the contribution of the proposed attention knowledge distillation loss in **AKD-GAN**, target facial attributes are applied much more strongly to the input face images. This is particularly true for *blond hair*, *rosy checks*, *mustache* and *pale skin* whose application was unsatisfactory in the lightweight student without attention knowledge distillation loss. Moreover, target facial attributes of *eyeglasses* and

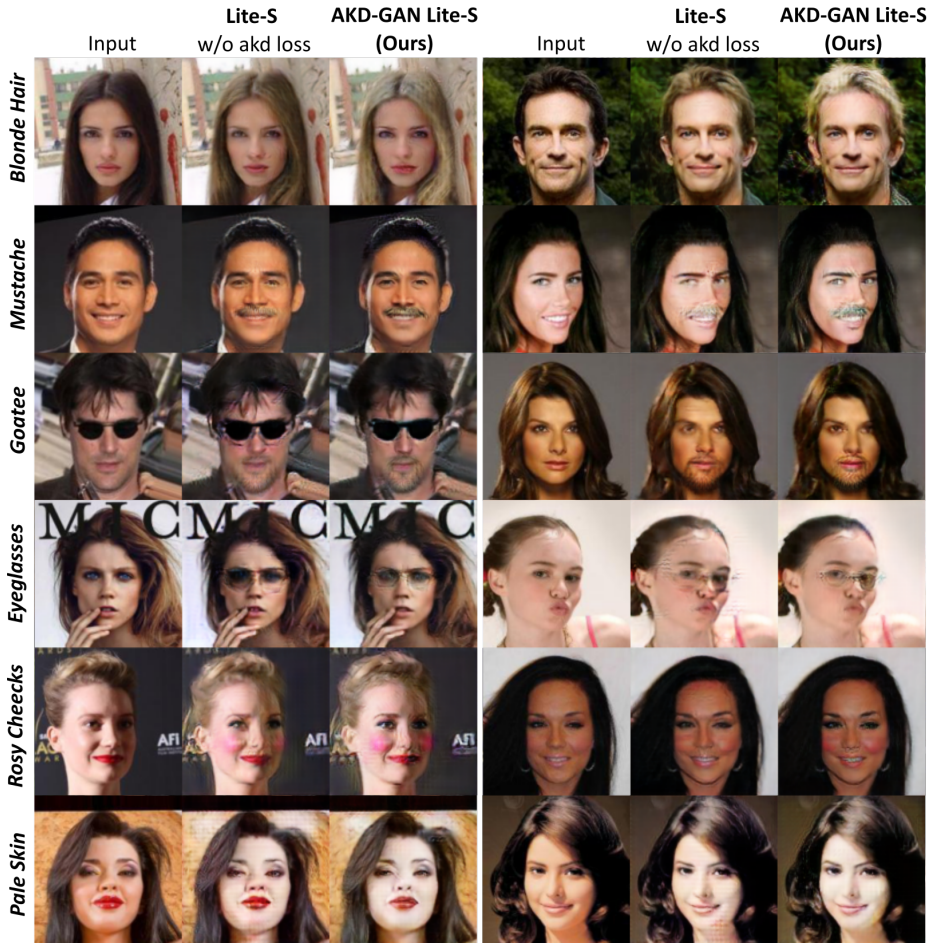

Figure 4.10: Collection of comparisons between qualitative results obtained from the lightweight student (***Lite-S***) without using the loss $\mathcal{L}_{akd}$ and results obtained when training (***Lite-S***) with the $\mathcal{L}_{akd}$ proposed in **AKD-GAN**.

heavy makeup are applied more convincingly to the input images. Finally, some undesired changes that can happen during the translation of certain facial attributes are less frequent. More specifically, when translating some facial attributes related to one particular gender, *i.e. goatee*, it can happen that gender is also translated in the output image, while, as discussed in Sec. 4.4.5, this effect is greatly reduced thanks to attention knowledge distillation.

Table 4.6: Classification results and overall FID scores over different facial attributes. The bold results represent the best results between the lightweight student network (***Lite-S***) trained without and with attention distillation loss, respectively.

Methods	Blond Hair ↑	Mustache ↑	Goatee ↑	Eyeglasses ↑	Rosy Cheeks ↑	Pale Skin ↑	Heavy Makeup ↑	Mean ↑	Overall FID ↓
Lite-S w/o akd loss (Baseline)	75.46%	24.71%	**51.38%**	56.48%	30.59%	67.14%	94.78%	57.22%	21.48
***Lite-S* with akd loss (Ours)**	**77.81%**	**24.87%**	48.46%	**62.65%**	**35.40%**	**70.90%**	**96.25%**	**59.47%**	**21.02**

Quantitative Results. Quantitative results are shown in Table 4.6. First of all, the classification accuracy of synthesized face images of each facial attribute is calculated using the pre-trained deep attribute classifier. Then, we also calculated the overall FID score for the synthesized face images for image quality evaluation.

Looking at the classification results, the lightweight student model (***Lite-S***) trained with attention distillation loss in **AKD-GAN** outperforms the one trained without attention distillation loss demonstrating the effectiveness of our approach. Regardless, distilling the attention knowledge with attention maps has shown its advantages in improving the performance of a lightweight network without altering the network design, demonstrating that visual attention serves more purpose than just visualization and can be used during training with success.

Regarding the overall FID score, the lightweight student model trained with attention distillation loss shows a slight boost in visual quality over the one trained without attention distillation loss but with a superior ability in translating the facial attributes.

4.4.6 Pseudo-Attention Knowledge Distillation for Face Synthesis with Facial Attributes Translation

We present "pseudo"-attention knowledge distillation experimental results under different teacher-student designs in the proposed framework **AKD-GAN**.

"Pseudo"-attention Knowledge Distillation with the Teacher and Student: We first visualize attention maps for facial attribute translation from the models ($\boldsymbol{T^{pse}}$ and $\boldsymbol{T}$) trained for different attributes ($\mathbf{c^{pse}}$ and $\mathbf{c}$) in Fig. 4.11. The top row shows attention maps generated from the model trained on a set of facial attributes $\mathbf{c^{pse}}$ (*wearing hat* and *black hair* for instance), while the bottom row shows attention maps generated from another model trained on a set of facial attributes $\mathbf{c}$ (*bald* and *brown hair*). It is evident that attention maps (last column) of each pair of facial attributes (*wearing hat* vs. *bald* and *black hair* vs. *brown hair*) share some spatial features in the input face images. This provides a strong evidence that it is possible to distill knowledge by defining and using "pseudo"-attentions from a teacher network to a student network for learning a *new* set of facial attributes.

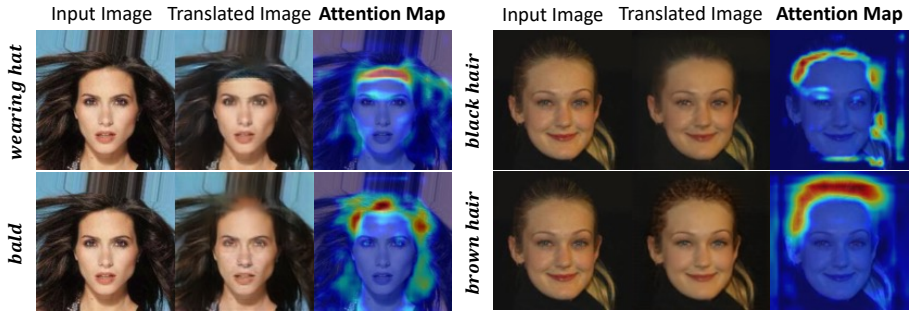

Figure 4.11: "Pseudo"-attention maps generated for facial attribute translation targeting two different sets of attributes.

We conducted experiments with **AKD-GAN** for facial attributes translation by using the proposed "pseudo"-attention knowledge distillation module. Firstly, we train one teacher model ($\boldsymbol{T^{pse}}$) to translate input face images to facial attributes ($\mathbf{c^{pse}}$) that are *black hair*, *arched eyebrows*, *big lips*, and *big nose*, etc. Secondly, we trained the student network ($\boldsymbol{S}$) with **AKD-GAN** to translate input face images to different facial attributes ($\mathbf{c}$) that are *brown hair*, *bushy eyebrows*,

wear lipstick and *pointy nose*, etc., with the proposed "pseudo"-attention knowledge distillation loss. The attention maps generated on the student network are the so-called "pseudo"-attentions and "pseudo"-attention knowledge distillation loss is calculated as a weak supervisory signals for training the student network.

Table 4.7: Classification accuracy and the overall FID scores over a group of different facial attributes using **AKD-GAN** with "pseudo"-attention knowledge distillation loss. The bold results represent the best results.

Methods	Teacher Attributes:	Brown Hair	Rosy Cheeks	Pointy Nose	Arched Eyebrows	Big Lips	Big Nose	Overall FID ↓
	Student Attributes:	Black Hair ↑	High Cheekbones ↑	Big Nose ↑	Bushy Eyebrows ↑	Wear Lipstick ↑	Pointy Nose ↑	
AKD-GAN (Ours)	**"pseudo"-attention knowledge distillation**	88.30%	**95.92%**	**94.66%**	**96.97%**	**97.60%**	**84.8%**	**14.34**
StarGAN [77] (Baseline)	No distillation	**94.28%**	94.92%	91.43%	95.6%	95.92%	82.44%	18.43

Quantitative Results. Quantitative results are presented in Table 4.7. Classification accuracy of synthesized face images of each attribute is calculated to evaluate the performance of the translation over face images with each target facial attributes and overall FID score is calculated to evaluate the quality of synthesized face images. From the results in Table 4.7, the proposed **AKD-GAN** trained with attention knowledge distillation loss using "pseudo"-attentions can consistently improve the translation of facial attributes and the quality of generated face images with target facial attributes.

It is observed that, when facial attributes are related to hair colors, *e.g.*, *blond hair*, *brown hair*, etc. our teacher model can be trained on just one of them and be used to distill attention knowledge for all the others. To further show that, in Table. 4.8, we use the proposed pseudo-attention knowledge distillation to train two additional experiments: (a) to distill attention knowledge from the teacher trained on facial attribute *"Blond_Hair"* to different attributes, *i.e.*, *"Blond_Hair"*, *"Brown_Hair"* and *"Gray_Hair"*, respectively. (b) do the same things as in (a) but training the

teacher using *"Black_Hair"*. It can be seen that "pseudo"-attention knowledge distillation module can consistently improve model performance by distilling attention knowledge from one facial attribute to a group of different attributes.

Table 4.8: Classification accuracy over different "hair color" attributes using **AKD-GAN** with attention knowledge distillation loss. The bold results represent the best results.

Methods	Teacher Attribute:	Blond Hair		
	Student Attributes:	Blond Hair ↑	Brown Hair ↑	Gray Hair ↑
AKD-GAN (Ours)	**"pseudo"-attention knowledge distillation**	**86.02%**	**88.59%**	**89.46%**
	Teacher Attribute:	Black Hair		
AKD-GAN (Ours)	**"pseudo"-attention knowledge distillation**	**84.89%**	**89.59%**	**77.33%**
StarGAN [77] (Baseline)	No distillation	80.94%	86.50%	76.94%

Qualitative Results. Sample synthesized face images are presented in Fig. 4.12. We can see that the results of the proposed **AKD-GAN** with "pseudo"-attention knowledge distillation loss can generate better facial attributes than the state-of-art method StarGAN [77]. This is particularly evident when the translation appears in only a small region of the input face image.

"Pseudo"-attention Knowledge Distillation with the Teacher and Lightweight Student: Following the similar experimental settings as in Sec. (4.4.5), the objective is to use "pseudo"-attentions defined in Sec. 4.3.3 to distill knowledge between a teacher $\boldsymbol{T}$ network targeting on a set of facial attributes and a lightweight student (***Lite-S***) network targeting on a different set of facial attributes. Firstly, we trained one teacher model ($\boldsymbol{T^{pse}}$) to translate input images to facial attributes ($\mathbf{c^{pse}}$) that are *black hair*, *blond hair*, *wearing hat* and *wavy hair*. Secondly, we trained the lightweight student network (***Lite-S***) with **AKD-GAN** to translate input images to different facial attributes ($\mathbf{c}$) that are *brown hair*, *gray hair*, *bald* and *straight hair* with the attention

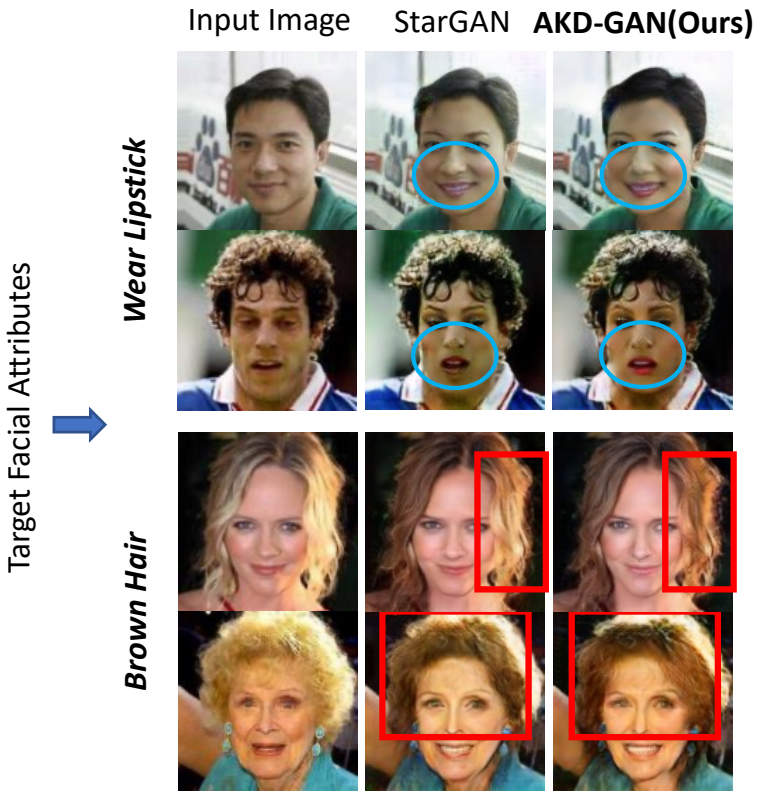

Figure 4.12: Qualitative results of AKD-GAN with "pseudo"-attention maps.

knowledge distillation loss. Finally, "pseudo"-attentions are used to distill knowledge between the teacher network and the lightweight student network (***Lite-S***).

Table 4.9: Classification results and the overall FID scores over different facial attributes for generated face images from the lightweight student network (***Lite-S***) trained using "pseudo"-attention distillation loss.

Methods \ Target Attributes:	Brown Hair ↑	Gray Hair ↑	Bald ↑	Straight Hair ↑	Avg. Accuracy ↑	Overall FID ↓
***Lite-S* w/o akd loss (Baseline)**	79.16%	32.46%	8.03%	56.48%	44.03%	27.02
***Lite-S* with akd loss (Ours)**	**82.77%**	**32.66%**	**12.43%**	**86.13%**	**53.5%**	**24.72**

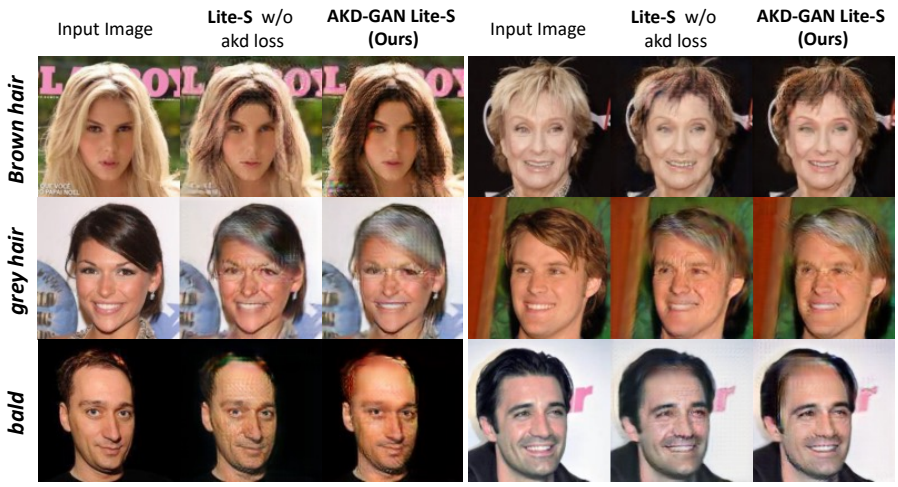

Figure 4.13: Qualitative results obtained from the lightweight student (***Lite-S***) trained with and without the proposed "pseudo"-attention distillation loss.

Table 4.10: Comparisons of classification accuracy over different human expressions using the proposed **AKD-GAN**.

Methods	Happy ↑	Angry ↑	Sad ↑	Contemptuous ↑	Disgusted ↑
StarGAN [77] (Baseline)	77.25%	71.37%	66.31%	84.00%	80.00%
AKD-GAN (Ours)	**79.31%**	**75.63%**	**73.81%**	**84.88%**	**86.75%**
	Neutral ↑	Fearful ↑	Surprised ↑	Mean ↑	
StarGAN [77] (Baseline)	67.93%	85.18%	81.56%	76.70%	
AKD-GAN (Ours)	**71.44%**	**86.88%**	**92.82%**	**81.44%**	

Qualitative Results. We present translated face images in Fig. 4.13. We can see that results of the lightweight student network (***Lite-S***) trained with the proposed attention distillation loss are more convincing than the outputs from the same model that does not employ attention distillation loss. This is particularly evident for facial attributes of *brown hair*, *grey hair* and *bald*.

Quantitative Results. Furthermore, the classification accuracy of translated face images of each attribute and the overall FID are calculated in Table 4.9. From results in Table 4.9, the lightweight student network (***Lite-S***) trained with attention distillation loss using "pseudo"-

attentions outperforms the network trained without it, which proves the effectiveness of our approach.

4.4.7 Attention Knowledge Distillation for Face Synthesis with Human Facial Expressions Translation

We conducted experiments by using the proposed method **AKD-GAN** for human facial expressions translation on the Radboud Faces Database (**RaFD**) [229] which consists of different human facial expressions. We follow the proposed experimental settings as in Sec. 4.4.5 and use two different designs for *teacher-student* and *teacher-lightweight-student*. We use classification accuracy to evaluate the performance of the proposed method for face synthesis with facial expressions translation and FID scores is not calculated as an evaluation metric since the test images in the dataset are too few. The first experiment is conducted on 8 facial expressions translation. Classification accuracy of synthesized face images with different human facial expressions is presented in Table 4.10. Improved classification accuracy can be observed in most facial expressions by using the proposed **AKD-GAN**, especially for *sad*, *disgusted*, and *surprised*, which proves the effectiveness of the proposed method on face synthesis with human facial expression translation.

Some qualitative results of human facial expression translation on *angry*, *disgusted*, *neutral*, and *surprised*, are presented in Fig. 4.14. We can see that better human facial expression translation images are obtained by using the proposed **AKD-GAN**, which validates the effectiveness of our method.

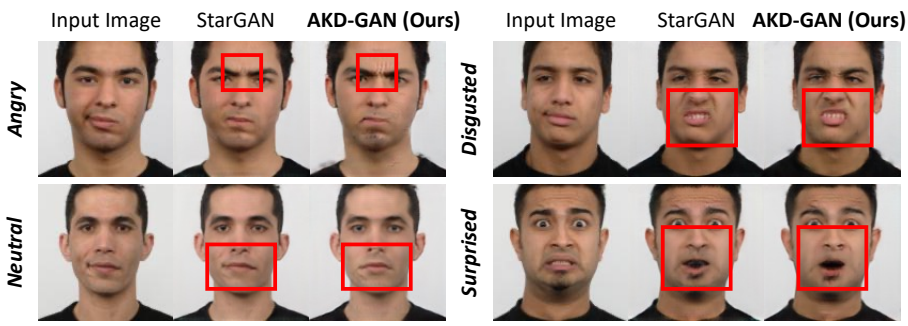

Figure 4.14: Qualitative results of human facial expression translation using **AKD-GAN**. *Note: better views are obtained with zooming in.*

Furthermore, we conducted experiments using lightweight student network (***Lite-S***). Classification results on 4 different facial expressions, *disgusted*, *fearful*, *happy* and *sad*, on the synthesized face images are presented in Table 4.11.

Table 4.11: Comparisons of classification results over 4 different human expressions using lightweight student network (***Lite-S***) in **AKD-GAN**.

Methods	Disgusted ↑	Fearful ↑	Happy ↑	Sad ↑	Mean ↑
Lite-S w/o akd loss	100.00%	95.31%	96.85%	70.57%	90.68%
***Lite-S* with akd loss (Ours)**	**100.00%**	**97.13%**	**97.91%**	**72.13%**	**91.79%**

4.4.8 Ablation Studies

In this sub-section, we explain the design choice of using last residual block for generating attention maps from two perspectives: empirical observations and additional ablation experiments. As introduced in Section 7.1, exploiting visual interpretations in image-to-image conditional adversarial networks is a fundamental step in order to improve upon them.

First, we investigate the architecture of a typical image-conditioned cGANs, *e.g.*, StarGAN. It consists of an encoder followed by residual blocks, then a decoder to decode features to images with high resolution. It is known that convolutional layers naturally retain spatial informa-

tion, so we can expect the last convolutional layers to have the best representation between high-level semantics and detailed spatial information. Therefore, in StarGAN, the encoder and residual blocks process the input images and target attributes step by step, then the neurons in convolutional layers of residual blocks encode semantic and spatial information of the input images and target attributes, outputting feature representations for the decoder. We streamline the attention generation by using the gradient information flowing into the last residual blocks of the generator for a particular facial attributes translation of interest.

Second, this design choice is supported by preliminary experimental observations shown in Fig. 4.2, where we present visual attention maps extracted from different residual blocks. The produced attention maps highlight the spatial information of features, where the generator focuses on a certain facial attribute and allow us to identify which layers in the generator are more devoted to the facial attributes translation task. The most refined attention can be seen in the last residual block (highlighted in red). This is a crucial observation for the development of our system, since it motivates us to derive the proposed attention knowledge distillation with the attention maps extracted from the last residual block. We argue that the attention maps extracted from previous layers are more noisy and deriving attention knowledge distillation using these attention maps would mislead the model training. The designed experiments given in previous sections and described below validate our proposed system.

Table 4.12: Ablation results by using our AKD-GAN with attention maps generated from the ***second last*** and ***last*** residual blocks for attention knowledge distillation during training.

Methods	Attention Knowledge Distillation	Interpretations	Blond Hair ↑	Goatee ↑	Eyeglasses ↑	Heavy Makeup ↑	Pale Skin ↑
AKD-GAN (Ours)	✓(second last block)	✓	81.22%	68.53%	98.35%	86.00%	83.11%
AKD-GAN (Ours)	✓(last block)	✓	**86.02%**	**74.45%**	**99.48%**	**91.47%**	**88.32%**

We present ablation experimental results on CelebA dataset in Table 4.12 by using the proposed **AKD-GAN** with the attention knowledge distillation but attention maps are generated from the ***second last*** residual blocks and compare them with the **AKD-GAN** trained with attention knowledge distillation and attention maps generated from the ***last*** residual blocks. From Table 4.12, it can be observed that using attention maps generated from the ***second last*** residual blocks for attention knowledge distillation gives worse performance than using attention maps generated from the last residual blocks. Given the attention maps shown in Fig. 4.2, we argue that the main reason for this is that the attention maps generated from the ***second last*** residual blocks contain inaccurate spatial information which would mislead the model training if these attention maps are used for attention knowledge distillation.

Chapter 5

Monoindoor++: Towards Better Practice of Self-supervised Monocular Depth Estimation for Indoor Environments

5.1 Introduction

Monocular depth estimation has been applied in a variety of 3D perceptual tasks, including autonomous driving, virtual reality (VR), and augmented reality (AR). Estimating the depth map plays an essential role in these applications, in helping to understand environments, plan agents' motions, reconstruct 3D scenes, etc. Existing supervised depth methods [114, 126] can achieve high performance, but they require the ground-truth depth data during the training which is often expensive and time-consuming to obtain by using depth sensors (*e.g.*, LiDAR). To this end, a number of recent work [130, 498, 141] have been focused on predicting the depth map from a single im-

age using self-supervised manners and they have shown advantages in scenarios where obtaining the ground-truth is not possible. In these self-supervised frameworks, photometric consistency between multiple views from monocular video sequences has been utilized as the main supervision for training models. Specifically, the recent work [141] has achieved significant success in estimating depth that is comparable to that by the supervised methods [152, 126]. For instance, on the KITTI dataset [132], the Monodepth2, proposed by Godard *et al.* [141], achieves an absolute relative depth error (AbsRel) of 10.6%, which is not far from the AbsRel of 7.2% by the DORN which is a supervised model proposed by Fu *et al.* [126]. However, most of these self-supervised depth prediction methods [130, 498, 141] are only evaluated on datasets of outdoor scenes such as KITTI, leaving their performance opaque for indoor environments. There are certainly ongoing efforts [495, 483, 31] which consider self-supervised monocular depth estimation for indoor environments, but their performance still trail far behind the one on the outdoor datasets by methods such as [130, 498, 141] or the supervised counterparts [126, 455] on indoor datasets. In this paper, we concentrate on estimating the depth map from a single image for indoor environments in a self-supervised manner which only requires monocular video sequences for training.

This paper investigates the performance discrepancies between the indoor and outdoor scenes and takes a step towards examining what makes indoor depth prediction more challenging than the outdoor case. We *first* identify that the scene depth range of indoor video sequences varies a lot more than in the outdoor and conjecture that this posits more difficulties for the depth network in inducing consistent depth cues across images from monocular videos, resulting in the worse performance on indoor datasets.

Our *second* observation is that the pose network, which is commonly used in self-supervised methods [498, 141], tends to have large errors in predicting rotational parts of relative camera poses. A similar finding has been presented in [502] where predicted poses have much higher rotational errors (*e.g.*, 10 times larger) than geometric SLAM [300] even when they use a recurrent neural network as the backbone to model the long-term dependency for pose estimation. We argue that this problem is not prominent on outdoor datasets, *i.e.*, KITTI, because the camera motions therein are mostly translational. However, frequent cameras rotations are inevitable in indoor monocular videos [376, 364] as these datasets are often captured by hand-held cameras or Micro Aerial Vehicles (MAVs). Thus, the inaccurate rotation prediction becomes detrimental to the self-supervised training of a depth model for indoor environments.

Our *third* conjecture is that the pose network in existing self-supervised methods is potentially suffering from insufficient cues to estimate relative cameras poses between color image pairs in different views. We argue that, rather than simply inducing camera poses based on color information of image pairs, encoding coordinates information can further improve the reliability of pose network in inferring geometric relations among changing views.

We propose **MonoIndoor++**, a self-supervised monocular depth estimation method tailored for indoor environments, giving special considerations for above problems. Our MonoIndoor++ consists of three novel modules: a *depth factorization* module, a *residual pose estimation* module, and a *coordinates convolutional encoding* module. In the depth factorization module, we factorize the depth map into a global depth scale (for the target image of the current view) and a relative depth map. The depth scale factor is separately predicted by an extra module (named as

transformer-based scale regression network) in parallel with the depth network which predicts a relative depth map. In such a way, the depth network has more model plasticity to adapt to the depth scale changes during training. We leverage the recent advances of transformer [110] in designing the scale regression network to predict the depth scale factor. In the residual pose estimation module, we mitigate the issue of inaccurate camera rotation prediction by performing residual pose estimation in addition to an initial large pose prediction. Such a residual approach leads to more accurate computation of the photometric loss [141], which in turn improves model training for the depth prediction. In the coordinates convolutional encoding module, we encode the coordinates information (x, y) explicitly and incorporate them with color information in the residual pose estimation module, expecting to provide additional cues for pose predictions, which further consolidates residual pose estimation model during training.

5.2 Related Work and Our Contributions

Much effort has been expended for the depth estimation in various environments. This paper addresses the self-supervised monocular depth estimation for indoor environments. In this section, we discuss the relevant work of depth estimation using both supervised and self-supervised methods.

5.2.1 Supervised Monocular Depth Estimation

The depth estimation problem was mostly solved by using supervised methods in early research. Saxena *et al.* [360] proposed the method to regress the depth from a single image by extracting superpixel features and using a Markov Random Field (MRF). Schonberger *et al.* [366]

presented a system for the joint estimation of depth and normal information with photometric and geometric priors. These methods employ traditional geometry-based methods. Eigen *et al.* [115] proposed the first deep-learning based method for monocular depth estimation using a multi-scale convolutional neural network (CNN). Later on, deep-learning based methods have shown significant progress on monocular depth estimation, specifically with massive ground-truth data during training the networks. One line of following work improves the performance of depth prediction by better network architecture design. Laina *et al.* [227] proposed an end-to-end fully convolutional architecture by encompassing the residual learning to predict accurate single-view depth maps given monocular images. Bhat *et al.* [28] proposed a transformer-based architecture block to adaptively estimate depth maps using a number of bins. Another line of work achieves improved depth estimation results by integrating more sophisticated training losses [236, 126, 455, 383, 46, 45]. Besides, a few methods [410, 394] proposed to use two networks, one for depth prediction and the other for motion, to mimic geometric Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) in a supervised framework. However, ground-truth depth maps with images are used to train these methods and obtaining ground-truth data is often expensive and time-consuming to capture. Some other methods then resorted to remedy this problem by generating pseudo ground-truth depth labels with traditional 3D reconstruction methods [249, 248], such as SfM [364] and SLAM [300, 395], or 3D movies [340]. Such methods have better capacity of generalization across different datasets, but cannot necessarily achieve the best performance for the dataset at hand. Some other ongoing efforts explore to improve robustness of supervised monocular depth estimation for zero-shot cross-dataset transfer. Ranftl *et al.* [340] proposed robust scale-and shift-invariant losses

for training the model using data from mixed dataset and testing on zero-shot datasets, and improved it further by integrating vision transformer in network design [339].

5.2.2 Self-Supervised Monocular Depth Estimation

Recently, significant progress has been made in self-supervised depth estimation as it does not require training with the ground-truth data. Garg *et al.* [130] proposed the first self-supervised method to train a CNN-based model for monocular depth estimation by using color consistency loss between stereo images. Zhou *et al.* [498] employed a depth network for depth estimation and a pose network to estimate relative camera poses between temporal frames, and used outputs to construct the photometric loss across temporal frames to train the model. Many follow-up methods then tried to propose new training loss terms to improve self-supervision for training models. Godard *et al.* [140] incorporated a left-right depth consistency loss for the stereo training. Bian *et al.* [29] put forth a temporal depth consistency loss to ensure predicted depth maps of neighbouring frames are consistent. Wang *et al.* [418] first observed the diminishing issue of the depth model during training and proposed a normalization method to counter this effect. Yin *et al.* [456] and Zou *et al.* [503] trained three networks (*i.e.*, one depth network, one pose network, and one extra flow network) jointly by enforcing cross-task consistency between optical flow and dense depth. Wang *et al.* [426] and Zou *et al.* [502] explored techniques to improve the performance of pose network and/or the depth network by leveraging recurrent neural networks, such as LSTMs, to model long-term dependency. Tiwari *et al.* [399] designed a self-improving loop with monocular SLAM and a self-supervised depth model [141] to improve the performance of each one. Among these recent advances, Monodepth2 [141] significantly improved the performance over previous methods via a set of techniques: a per-pixel minimum photometric loss to handle occlusions, an auto-masking

method to mask out static pixels, and a multi-scale depth estimation strategy to mitigate the texture-copying issue in depth. Watson *et al.* [437] proposed to use cost volume in the deep model and a new consistency loss calculated between the a teacher and a student model for self-supervision training. Unlike Monodepth2, this method showed its advantages in using multiple frames during the testing. We implement our self-supervised depth estimation framework based on Monodepth2, but make important changes in designing both the depth and the pose networks.

Most of the aforementioned methods were only evaluated on outdoor datasets such as KITTI. Recent ongoing efforts [495, 483, 30] focus on self-supervised depth estimation for indoor environments. Zhou *et al.* [495] first observed existing large rotations on most existing indoor datasets, and then used a pre-processing step to handle large rotational motions by removing all the image pairs with "pure rotation" and designed an optical-flow based training paradigm using the processed data. Zhao *et al.* [483] adopted a geometry-augmented strategy that solved for the depth via two-view triangulation and then used the triangulated depth as supervision for model training. Bian *et al.* [30, 31] theoretically studied the reasons behind the unsatisfying deep estimation performance in indoor environments and argued that "the rotation behaves as noise during training". They proposed a rectification step during the data pre-precessing to remove the rotation between consecutive frames and designed an auto-rectify network. We have an observation similar to [495, 30] and [31] that large rotations cause difficulties for training the network. However, we take a different strategy. Instead of removing rotations from training data during the data pre-processing, we progressively estimate camera poses in rotations and translations via a novel residual pose module in an end-to-end manner, and we validate the effectiveness of the proposed method in predicting improved depth on a variety of indoor benchmark datasets.

5.2.3 Transformer

We leverage the transformer in designing our scale regression network inspired by the recent advances [431, 413, 110] of the attention mechanism. Self-attention in the transformer was first used successfully in natural lanuage processing (NLP) to model long-term dependencies. Wang *et al.* [431] proposed a non-local operations for computer vision tasks. Recently, self-attention and its variants have been widely used in transformer networks for high-level visions tasks such as image classification [110] and semantic segmentation [275, 273].

5.2.4 Coordinates Encoding

Convolutional neural networks (CNNs) have achieved significant success at many tasks, and it can be complemented with specialized layers for certain usage. For instance, detection models like Faster R-CNN [345] make use of layers to compute coordinate transforms. Jaderberg *et al.* [186] proposed a spatial Transformer module that can be included into a standard CNN model to provide spatial transformation capabilities. Qi *et al.* [331, 332] designed the PointNet which took a set of 3D points represented as (x, y, z) coordinates as well as extra color features for 3D classification and segmentation. Recently, coordinates encoding has been widely used in vision transformers [110, 275, 273] and neural radiance fields (NERF) representations [297, 312, 389]. Vision transformers [110] take 2D images as the input, reshape the image into a sequence of flattened 2D patches and then employ self-attention blocks for image classification, detection and segmentation. Position embeddings are added to the patch embeddings as a standard processing step to retain positional information. Ben Mildenhall *et al.* [297] proposed a method which took a 3D location (x, y, z) and 2D viewing direction (θ, ϕ) as the input for scene synthesis. Unlike in vi-

sion transformers where positional encoding is utilized to provide discrete positions of tokens in the sequence, in NERF, positional functions are used to map continuous input coordinates into a higher dimensional space for high frequency approximations. Liu *et al.* [266] defined the *CoordConv* operation to provide extra coordinates information as part of input channels to convolutional filters for the convolutional neural networks. Most of pose networks in monocular depth estimation pipelines [140, 141] simply take two consecutive frames as the input and outputs relative camera poses. We argue such designs infer rotational and translational relations by only focusing on photometric cues, but ignoring explicit coordinates cues. In our work, we leverage coordinates encoding in the proposed residual pose network.

5.2.5 Monocular Depth Estimation for the Circuits and Systems for Video Technology

Depth estimation is one of the most fundamental tasks in computer vision for the circuits and systems for video technology, and it has made great progress in recent years. Using deep learning techniques, efforts have been made to estimate dense depth maps, given input images, in a supervised manner. Cao *et al.* [45] formulated the estimation of depth as a pixelwise classification problem with a fully convolutional depth residual network. Song *et al.* [383] incorporated the idea of the Laplacian pyramid into the depth decoder. During the training, depth encoder features are fed into different streams which are predefined by the decomposition of the Laplacian pyramid for outputing the final depth map. Rather than employing a fully supervised approach for monocular depth estimation, Tian *et al.* [398] used quadtree constraint for calculating the photometric and depth loss during the training of a depth model. It leveraged the sparse depth information as a part of the input during semi-supervised training. To enable fully self-supervised training based on the

standard framework designed by Zhou *et al.* [498], Chen *et al.* [63] incorporated additional losses derived from SURF features and mapped point clouds. However, different from [383, 45] which concentrated on supervised depth estimation, Tian *et al.* [398] and Chen *et al.* [63] leveraged sparse depth information from the visual odometry system and explored additional supervisions for self-supervised monocular training but they still suffered from unsatisfactory performance (AbsRel of 16.5% on NYUv2). In this paper, we proposed a novel framework, **MonoIndoor++**, with three new modules, a depth factorization module, a residual pose estimation module, and a coordinates convolutional encoding module, which target on solving existing problems, rapid scale changes in indoor environments, inaccurate camera rotation prediction issue and missing coordinates cues in inducing relative camera poses, for self-supervised monocular depth estimation in indoor environments. Our model can be trained with standard photometric loss derived from self-supervision and has established state-of-the-art (SOTA) performance on a wide-range of challenging benchmark indoor datasets.

5.2.6 Contributions of this Chapter

- We propose a novel depth factorization module with a transformer-based scale regression network to estimate a global depth scale factor, which helps the depth network adapt to the rapid scale changes for indoor environments during model training.

- We propose a novel residual pose estimation module that mitigates the inaccurate camera rotation prediction issue in the pose network and in turn significantly improves monocular depth estimation performance.

- We incorporate coordinates convolutional encoding in the proposed residual pose estimation module to leverage coordinates cues in inducing relative camera poses.

- We demonstrate the state-of-the-art performance of self-supervised monocular depth prediction on a wide-variety of publicly available indoor datasets, *i.e.*, NYUv2 [376], EuRoC MAV [42], ScanNet [91] and 7-Scenes [373].

5.3 Technical Approach

In this section, we present detailed descriptions of performing self-supervised depth estimation using the proposed **MonoIndoor++**. Specifically, we first give an overview of the standard framework for the self-supervised depth estimation. Then, we describe three core components including depth factorization, residual pose and coordinates convolution modules, respectively.

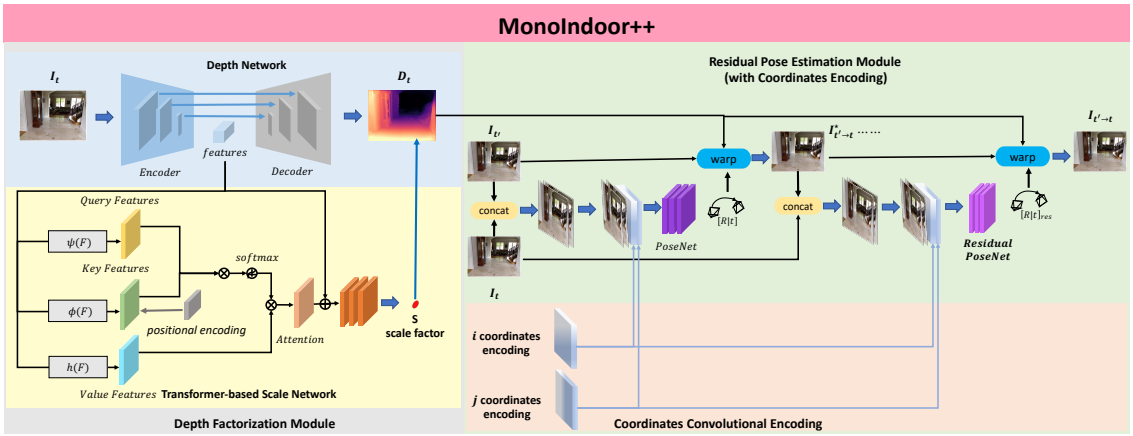

Figure 5.1: Overview of the proposed **MonoIndoor++**. **Depth Factorization Module**: We use an encoder-decoder based depth network to predict a relative depth map and a transformer-based scale network to estimate a global scale factor. **Residual Pose Estimation Module**: We use a pose network to predict an initial camera pose of a pair of frames and residual pose network to iteratively predict residual camera poses based on the predicted initial pose. **Coordinates Convolutional Encoding**: We encode coordinates information along with the concatenated color image pairs as the input to the pose network and residual pose network for predicting relative camera poses.

5.3.1 Self-Supervised Monocular Depth Estimation

Self-supervised monocular depth estimation is considered as a novel view-synthesis problem which is defined in [498, 141, 502]. This key idea is to train a model to predict the target image from different viewpoints of source images. The image synthesis is achieved by using the depth map as the bridging variable between the depth network and pose network. Both the depth map of the target image and the estimated relative camera pose between a pair of target and source images are required to train such systems. Specifically, the depth network predicts a dense depth map D_t given a target image I_t as the input. The pose network takes a target image I_t and a source image $I_{t'}$ from another view and estimates a relative camera pose $T_{t\to t'}$ from the target to the source. The depth network and pose network are optimized jointly with the photometric reprojection loss which can then be constructed as follows:

$$\mathcal{L}_A = \sum_{t'} \rho(I_t, I_{t'\to t}), \tag{5.1}$$

and

$$I_{t'\to t} = I_{t'}\langle proj(D_t, T_{t\to t'}, K)\rangle, \tag{5.2}$$

where ρ denotes the photometric reconstruction error [498, 141]. It is a weighted combination of the L1 and Structured SIMilarity (SSIM) loss defined as

$$\rho(I_t, I_{t'\to t}) = \frac{\alpha}{2}\big(1 - \texttt{SSIM}(I_t, I_{t'\to t})\big) + (1-\alpha)\|I_t, I_{t'\to t}\|_1. \tag{5.3}$$

$I_{t'\to t}$ is the source image warped to the target coordinate frame based on the depth of the target image which is the output from the depth network. $proj()$ is the transformation function to map image coordinated p_t from the target image to its $p_{t'}$ on the source image following

$$p_{t'} \sim KT_{t\to t'}D_t(p_t)K^{-1}p_t, \tag{5.4}$$

and $\langle\cdot\rangle$ is the bilinear sampling operator which is locally sub-differentiable.

In addition, an edge-ware smoothness term is normally employed during training which can be written as

$$\mathcal{L}_s = |\partial_x d_t^*|e^{-|\partial_x I_t|} + |\partial_y d_t^*|e^{-|\partial_y I_t|}, \tag{5.5}$$

where $d_t^* = d/\bar{d}_t$ is the mean-normalized inverse depth from [418].

Further, inspired by [29], we incorporate the depth consistency loss to enforce the predicted depth maps across the target frame and neighbouring source frames to be consistent during the training. We first warp the predicted depth map $D_{t'}$ of the source image $I_{t'}$ by Equation (5.2) to generate $D_{t'\to t}$, which is a corresponding depth map in the coordinate system of the source image. We then transform $D_{t'\to t}$ to the coordinate system of the target image via Equation (5.4) to produce a synthesized target depth map $\widetilde{D}_{t'\to t}$. The depth consistency loss can be written as

$$\mathcal{L}_d = \frac{|D_t - \widetilde{D}_{t'\to t}|}{D_t + \widetilde{D}_{t'\to t}}. \tag{5.6}$$

Thus, the overall objective to train the model is

$$\mathcal{L} = \mathcal{L}_A + \tau\mathcal{L}_s + \gamma\mathcal{L}_d, \tag{5.7}$$

where τ and γ are the weights for the edge-aware smoothness loss and the depth consistency loss respectively.

As discussed in Section 5.1, existing self-supervised monocular depth estimation models have been used widely in producing competitive depth maps on datasets collected in outdoor environments, *e.g.*, autonomous driving scenes. However, simply using these methods [141] still suffer

from worse performance in indoor environments, especially compared with fully-supervised depth prediction methods. We argue that the main challenges in indoor environments come from the fact that i) the depth range changes a lot and ii) indoor sequences captured in existing public indoor datasets, *e.g.*, EuRoC MAV [42] and NYUv2 [376], contain regular rotational motions which are difficult to predict. To handle these issues, we propose **MonoIndoor++**, a self-supervised monocular depth estimation framework, as shown in Figure 6.3, to enable improved predicted depth quality in indoor environments. The framework takes as input a single color image and outputs a depth map via our MonoIndoor++ which consists of two core parts: a depth factorization module with a transformer-based scale regression network and a residual pose estimation module. In addition, when designing the residual pose estimation, we incorporate coordinates convolutional operations to encode coordinates information along with color information as input channels explicitly. The details of our main contributions are presented in the following sections.

5.3.2 Depth Factorization Module

Our depth factorization module consists of a depth prediction network and a transformer-based scale regression network.

Depth Prediction Network: The backbone model of our depth prediction network is based on Monodepth2 [141], which employs an auto-encoder structure with skip connections between the encoder and the decoder. The depth encoder learns a feature representation given a color image I as input. The decoder takes features from the encoder as the input and outputs relative depth map prediction. In the decoder, a sigmoid activation function is used to process features from the last convolutional layers and a linear scaling function is utilized to obtain the final up-to-the-scale

depth prediction, which can be written as follows,

$$d = 1/(a\sigma + b), \tag{5.8}$$

where σ is the outputs after the sigmoid function, a and b are specified to constrain the depth map D within a certain depth range. a and b are pre-defined as a minimum depth value and a maximum depth value empirically according to a known environment. For instance, on the KITTI dataset [132] which is collected in outdoor scenes, a is chosen as 0.1 and b as 100. The reason for setting a and b as these fixed values is that the depth range is consistent across the video sequences when the camera always sees the sky at the far point. However, it is observed that this setting is not valid for most indoor environments. For instance, on the NYUv2 dataset [376] which include various indoor scenes, *e.g.*, office, kitchen, *etc.*, the depth range varies significantly as scene changes. Specifically, the depth range in a bathroom (*e.g.*, 0.1m$\sim$3m) can be very different from the one in a lobby (*e.g.*, 0.1m$\sim$10m). We argue that pre-setting depth range will act as an inaccurate guidance that is harmful for the model to capture accurate depth scales in training models. This is especially true when there are rapid scale changes, which are commonly observed on datasets [376, 42, 91] in indoor scenes. Therefore, to mitigate this problem, our depth factorization module learns a disentangled representation in the form of a relative depth map and a global scale factor. The relative depth map is obtained by the depth prediction network aforementioned and a global scale factor is outputted by a transformer-based scale regression network which is introduced in the next subsection.

Transformer-based Scale Regression Network: We propose a transformer-based scale regression network (see Figure 6.3) as a new branch which takes as input a color image and outputs its corresponding global scale factor. Our intuition is that the global scale factor can be informed by certain areas (*e.g.*, the far point) in the images, and we propose to use a transformer block to learn

the global scale factor. Our expectation is that the network can be guided to pay more attention to a certain area which is informative to induce the depth scale factor of the target image of the current view in a scene.

The proposed transformer-based scale regression network takes the feature representations $\mathcal{F} \in \mathbb{R}^{D \times H \times W}$ learnt from the input image as the input and outputs the corresponding global scale factor, where D is dimension, H and W are the height and width of the feature map. Specifically, we project input features $\mathcal{F} \in \mathbb{R}^{D \times H \times W}$ to the query, the key and the value output, which are defined as

$$\begin{aligned} \psi(\mathcal{F}) &= \mathbf{W}_{\psi}\mathcal{F}, \\ \phi(\mathcal{F}) &= \mathbf{W}_{\phi}\mathcal{F}, \\ h(\mathcal{F}) &= \mathbf{W}_{h}\mathcal{F}, \end{aligned} \tag{5.9}$$

where $\mathbf{W}_{\psi}$, $\mathbf{W}_{\phi}$ and $\mathbf{W}_{h}$ are parameters to be learnt. The query and key values are then combined using the function $\mathcal{G}_{\mathcal{F}} = \texttt{softmax}(\mathcal{F}^{T}\mathbf{W}_{\psi}^{T}\mathbf{W}_{\phi}\mathcal{F})h(\mathcal{F})$, giving the learnt self-attentions as $\mathcal{G}_{\mathcal{F}}$. Finally, the $\mathcal{G}_{\mathcal{F}}$ and the input feature representation $\mathcal{F}$ jointly contribute to the output $\mathcal{S}_{\mathcal{F}}$ by using

$$\mathcal{S}_{\mathcal{F}} = \mathbf{W}_{\mathcal{S}_{\mathcal{F}}}\mathcal{G}_{\mathcal{F}} + \mathcal{F}. \tag{5.10}$$

Once we obtain $\mathcal{S}_{\mathcal{F}}$, we apply three residual blocks including two convolutional layers in each, followed by three fully-connected layers with dropout layers in-between, to output the global scale factor S for the target image of current view. We also use a 2D relative positional encoding [21] in calculating attentions with considerations of relative positional information of key features.

Probabilistic Scale Regression Head: The proposed transformer-based scale regression network is designed is to predict a single positive number given the input high-dimensional feature map $\mathcal{F} \in \mathbb{R}^{D \times H \times W}$. Inspired by the stereo matching work [56], we propose to use a probabilistic

scale regression head to estimate the continuous value for scale factor. Specifically, given a maximum bound that the global scale factor is within, instead of outputting a single number directly, we first output a number of scale values $\widetilde{S}$ as the predictions of each scale s and then calculate the probability of s via the softmax operation $\texttt{softmax}(\cdot)$. Finally, the predicted global scale S is calculated as the sum of each scale s weighted by its probability of predicted values as

$$S = \sum_{s=0}^{D_{max}} s \times \texttt{softmax}(\widetilde{S}). \tag{5.11}$$

Thus, the probabilistic scale regression head enables us to resolve regression problem smoothly with a probabilistic classification-based strategy (see Section 5.4.5 for ablation results).

5.3.3 Residual Pose Estimation

The principle of self-supervised monocular depth estimation is built upon the novel view synthesis, which requires both accurate depth maps from the depth network and camera poses from the pose network. Estimating accurate relative camera poses is important for calculating photometric reprojection loss to train the model because inaccurate camera poses might lead to wrong correspondences between the pixels in the target and source images, positing problems in predicting accurate depth maps. A standalone "PoseNet" is widely used in existing methods [141] to take two images as the input and to estimate the 6 Degrees-of-Freedom (DoF) relative camera poses. On datasets in outdoor environments (*e.g.*, autonomous driving scenes like KITTI), we argue that the relative camera poses are fairly simple because the cars which are used to collect video data are mostly moving forward with large translations but minor rotations. This means that pose estimation is normally less challenging for the pose network. In contrast, in indoor environments, the video sequences in widely-used datasets [376] are typically recorded with hand-held devices (*e.g.*,

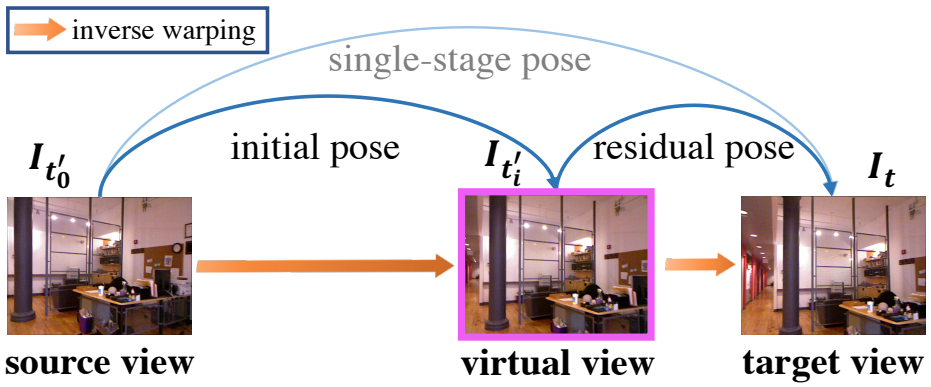

Figure 5.2: Residual Pose Estimation. A single-stage pose can be decomposed into an *initial pose* and a *residual pose* by virtual view synthesis.

Kinect), so there are more complicated ego-motions involved as well as much larger rotational motions. Thus, it is relatively more difficult for the pose network to learn to predict accurate relative camera poses.

To better mitigate the aforementioned issues, existing methods [495, 30] concentrate on "removing" or "reducing" rotational components in camera poses during data preprocessing and train their models using preprocessed data. In this work, we argue these preprocessing techniques are not flexible in end-to-end training pipelines, instead, we propose a residual pose estimation module to learn the relative camera pose between the target and source images from different views in an iterative manner (see Figure 5.2 for core ideas).

Our residual pose module consists of a standard pose network and a residual pose network. In the first stage, the pose network takes a target image I_t and a source image $I_{t'_0}$ as input and predicts an initial camera pose $T_{t'_0 \to t}$, where the subscript 0 in t'_0 indicates that no transformation is applied over the source image yet. Then Equation (5.2) is used to bilinearly sample from the source image, reconstructing a warped target image $I_{t'_0 \to t}$ of a virtual view which is expected to be the same as the

target image I_t if the correspondences are solved accurately. However, it will not be the case due to inaccurate pose prediction. The transformation for this warping operation is defined as

$$I_{t'_0 \to t} = I_{t'} \langle proj(D_t, T^{-1}_{t'_0 \to t}, K) \rangle. \tag{5.12}$$

Next, we propose a residual pose network (see ***ResidualPoseNet*** in Figure 6.3) which takes the target image and the synthesized target image of a virtual view ($I_{t'_0 \to t}$) as input and outputs a residual camera pose $T^{res}_{(t'_0 \to t) \to t}$, representing the camera pose of the synthesized image $I_{t'_0 \to t}$ with respect to the target image I_t. Then, we bilinearly sample from the synthesized image as

$$I_{(t'_0 \to t) \to t} = I_{t'_0 \to t} \langle proj(D_t, T^{res\ -1}_{(t'_0 \to t) \to t}, K) \rangle. \tag{5.13}$$

Once a new synthesized image of a virtual view is obtained, we can continue to estimate the residual camera poses for next view synthesis operation.

We define the general form of Equation (5.13) as

$$I_{t'_i \to t} = I_{t'_i} \langle proj(D_t, T^{res-1}_{t'_i \to t}, K) \rangle, i = 0, 1, \cdots . \tag{5.14}$$

by replacing the subscript $t'_0 \to t$ with t'_1 to indicate that one warping transformation is applied, and similarly for the i^{th} transformation.

To this end, after multiple residual poses are estimated, the camera pose of source image I'_t with respect to the target image I_t can be written as $T_{t \to t'} = T^{-1}_{t' \to t}$ where

$$T_{t' \to t} = \prod_i T_{t'_i \to t}, i = \cdots, k, \cdots, 1, 0 \ . \tag{5.15}$$

By iteratively estimating residual poses using a pose network and a residual pose network, we expect to obtain more accurate camera pose compared with the pose predicted from a single-stage pose network, so that a more accurate photometric reprojection loss can be built up for better depth prediction during the model training.

5.3.4 Coordinates Convolutional Encoding

For self-supervised monocular depth estimation, most of existing methods are designed to induce relative camera poses given a pair of color images. In this work, we propose to incorporate coordinates information as a part of input channels along with the color information explicitly to provide additional coordinates cues for pose estimation.

We extend standard convolutional layers to coordinates convolutional layers by initializing extra channels to process coordinates information which is concatenated channel-wise to the input representations (see **Coordinates Convolutional Encoding** in Figure 6.3). Given a pair of 2D images, we encode two coordinates x, y with color information (r, g, b), resulting in the 8-channels input as $(r_1, g_1, b_1, r_2, g_2, b_2, i, j)$ where (r_1, g_1, b_1) and (r_2, g_2, b_2) are rgb values of color images, respectively. The i coordinate channel is an $h \times w$ rank-1 matrix with its first row filled with 0's, its second row with 1's, its third with 2's, etc. The j coordinate channel is similar, but with columns filled in with constant values instead of rows. A linear scaling operation is applied over both i and j coordinate values to encode them in the range $[-1, 1]$. We adopt coordinates convolutional layers [266] in the residual pose estimation module to process 8-channels input for iterative pose estimation, and the pose estimation can be written as follows:

$$T_{t \to t'} = RPModule(\mathbf{\Omega}; Concat(I_t, I_{t'}, i, j)) \tag{5.16}$$

where $RPModule$ is the proposed pose estimation module, Ω is the parameters of the module which are to be optimized.

5.4 Experimental Results

5.4.1 Implementation Details

We implement our model using PyTorch [315]. In the depth factorization module, we use the same depth network as in Monodepth2 [141]; for the transformer-based scale regression network, we use a transformer module followed by two basic residual blocks and then three fully-connected layers with a dropout layer in-between. The dropout rate is empirically set to 0.5. In the residual pose module, we let the residual pose networks use a common architecture as in Monodepth2 [141] which consists of a shared pose encoder and an independent pose regressor. In the coordinates encoding module, 2D coordinates information (i, j) are directly concatenated with (r, g, b) channels of color images as the input and the convolutional layers are initialized with ImageNet-pretrained weights. Each experiment is trained for 40 epochs using the Adam [211] optimizer and the learning rate is set to 10^{-4} for the first 20 epochs and it drops to 10^{-5} for remaining epochs. The smoothness term τ is set as 0.001. The consistency term γ are set as 0.1 for EuRoC MAV dataset, 0.035 for NYUv2, ScanNet and 7-Scenes datasets, respectively.

5.4.2 Datasets

NYUv2 [376]: The NYUv2 depth dataset contains 464 indoor video sequences which are captured by a hand-held Microsoft Kinect RGB-D camera. The dataset is widely used as a challenging benchmark for depth prediction. The resolution of videos is 640$\times$ 480. Images are rectified with provided camera intrinsics to remove image distortion. We use the official training and validation

splits which include 302 and 33 sequences, respectively. We use officially provided 654 images with dense labelled depth maps for testing. During training, images are resized to 320×256.

EuRoC MAV [42]: The EuRoC MAV Dataset contains 11 video sequences captured in two main scenes, a machine hall and a vicon room. Sequences are categorized as *easy*, *medium* and *difficult* according to the varying illumination and camera motions. For the training, we use three sequences of "Machine hall" (MH_01, MH_02, MH_04) and two sequences of "Vicon room" (V1_01 and V1_02). Images are rectified with provided camera intrinsics to remove image distortion. During training, images are resized to 512×256. We use the Vicon room sequence V1_03, V2_01, V2_02 and V2_03 for testing where the ground-truth depths are generated by projecting Vicon 3D scans onto the image planes. During training, images are resized to 512×256. In addition, we use V2_01 for ablation studies (see Section 5.4.5 and Section 5.4.5).

ScanNet [91]: The ScanNet dataset contains RGB-D videos of 1513 indoor scenes, which is captured by handheld devices. The dataset is annotated with 3D camera poses and instance-level semantic segmentations and is widely on several 3D scene understanding tasks, including 3D object classification, semantic voxel labeling, and CAD model retrieval. We use officially released train-validation-test splits. The resolution of color images is 1296×968. During training, images are resized to 320×256.

7-Scenes [373]: 7-Scenes dataset contains a number of video sequences captured in 7 different indoor scenes, *i.e.*, *office*, *stairs*, etc. Each scene contains 500-1000 frames. All scenes are recorded using a handheld Kinect RGB-D camera at the resolution of 640×480. We use the official train-test split. During training, images are resized to 320×256.

5.4.3 Evaluation Metrics

We use both error metrics and accuracy metrics proposed in [115] for evaluation on all datasets, which include the mean absolute relative error (AbsRel), root mean squared error (RMS) and the accuracy under threshold ($\delta_i < 1.25^i, i = 1, 2, 3$). Following previous self-supervised depth estimation methods [141, 483, 31], we multiply the predicted depth maps by a scalar that matches the median with that of the ground-truth because self-supervised monocular methods cannot recover the metric scale. The predicted depths are capped at 10m in all indoor datasets except the EuRoC MAV dataset which one is set as 20m because it contains "Machine hall" scenes with observed large depth scale.

5.4.4 Experimental Results

Results on NYUv2 Depth Dataset: In this sub-section, we evaluate our **MonoIndoor++** on the NYUv2 depth dataset [376]. Following [483, 31], the raw dataset is firstly downsampled 10 times along the temporal dimension to remove redundant frames, resulting in $\sim 20K$ images for training.

Quantitative Results Table 5.1 presents the quantitative results of our model **MonoIndoor++** and both SOTA supervised and self-supervised methods on NYUv2. It shows that our model outperforms all previous self-supervised SOTA methods [141, 29, 460, 31], reaching the best results across all metrics. Specifically, our method improves monocular depth prediction performance significantly by reducing AbsRel to 13.2% and increasing δ_1 to 83.4%. Besides, compared with the recent self-supervised methods by Bian *et al.* [30, 31] which concentrating on removing rotations via "rectification" as a data preprocessing step, our method gives the better performance

without additional data preprocessing. It is noted that NYUv2 is very challenging and many previous self-supervised methods [456] fail to get satisfactory results. In addition to that, our model outperforms a group of supervised methods [262, 226, 114, 53, 236] and closes the performance gap between the self-supervised methods and fully-supervised methods [227, 126]. When compared with our preliminary work [187], our method can consistently improve depth estimation performance on all metrics, especially the δ_1, which is 83.4% and is better that these fully-supervised methods [227, 126]. Ablation studies for the effectiveness of each core module on NYUv2 are presented in Section 5.4.5 and the ablation results of design choices for the coordinates convolutional encoding are shown in Section 5.4.5.

Table 5.1: Comparison of our method with existing supervised and self-supervised methods on NYUv2 [376]. Best results among supervised and self-supervised methods are in **bold**.

Methods	Supervision	Error Metric ↓		Accuracy Metric ↑		
		AbsRel	RMS	δ_1	δ_2	δ_3
Make3D [360]	✓	0.349	1.214	0.447	0.745	0.897
Depth Transfer [202]	✓	0.349	1.210	-	-	-
Liu *et al.* [262]	✓	0.335	1.060	-	-	-
Ladicky *et al.* [226]	✓	-	-	0.542	0.829	0.941
Li *et al.* [232]	✓	0.232	0.821	0.621	0.886	0.968
Roy *et al.* [351]	✓	0.187	0.744	-	-	
Liu *et al.* [256]	✓	0.213	0.759	0.650	0.906	0.976
Wang *et al.* [425]	✓	0.220	0.745	0.605	0.890	0.970
Eigen *et al.* [114]	✓	0.158	0.641	0.769	0.950	0.988
Chakrabarti *et al.* [53]	✓	0.149	0.620	0.806	0.958	0.987
Laina *et al.* [227]	✓	0.127	0.573	0.811	0.953	0.988
Li *et al.* [236]	✓	0.143	0.635	0.788	0.958	0.991
DORN [126]	✓	0.115	0.509	0.828	0.965	0.992
Ranftl *et al.* [339]	✓	0.110	**0.357**	0.904	**0.988**	0.994
VNL [455]	✓	0.108	0.416	0.875	0.976	0.994
Bhat *et al.* [28]	✓	0.103	0.364	**0.903**	0.984	**0.997**
Fang *et al.* [121]	✓	**0.101**	0.412	0.868	0.958	0.986
Zhou *et al.* [495]	✗	0.208	0.712	0.674	0.900	0.968
Zhao *et al.* [483]	✗	0.189	0.686	0.701	0.912	0.978
Monodepth2 [141]	✗	0.160	0.601	0.767	0.949	0.988
SC-Depth [29]	✗	0.159	0.608	0.772	0.939	0.982
P^{2}Net (3-frame) [460]	✗	0.159	0.599	0.772	0.942	0.984
P^{2}Net (5-frame) [460]	✗	0.147	0.553	0.801	0.951	0.987
Bian *et al.* [30]	✗	0.147	0.536	0.804	0.950	0.986
Bian *et al.* [31]	✗	0.138	0.532	0.820	0.956	0.989
Monodepth2 [141] (Baseline)	✗	0.160	0.601	0.767	0.949	0.988
MonoIndoor [187] (Our ICCV21)	✗	0.134	0.526	0.823	0.958	0.989
MonoIndoor++ (Ours)	✗	**0.132**	**0.517**	**0.834**	**0.961**	**0.990**

Qualitative Results Figure 5.3 visualizes the predicted depth maps on NYUv2. Compared with the results from the baseline method Monodepth2 [141] and recent work [31], depth

maps predicted from our model (**MonoIndoor++**) are more precise and closer to the ground-truth. For instance, looking at the fourth column in the first row, the depth in the region of ***cabinet*** predicted from our model is much sharper and cleaner, being close to the ground-truth (the last column). These qualitative results are consistent with our quantitative results in Table 5.1.

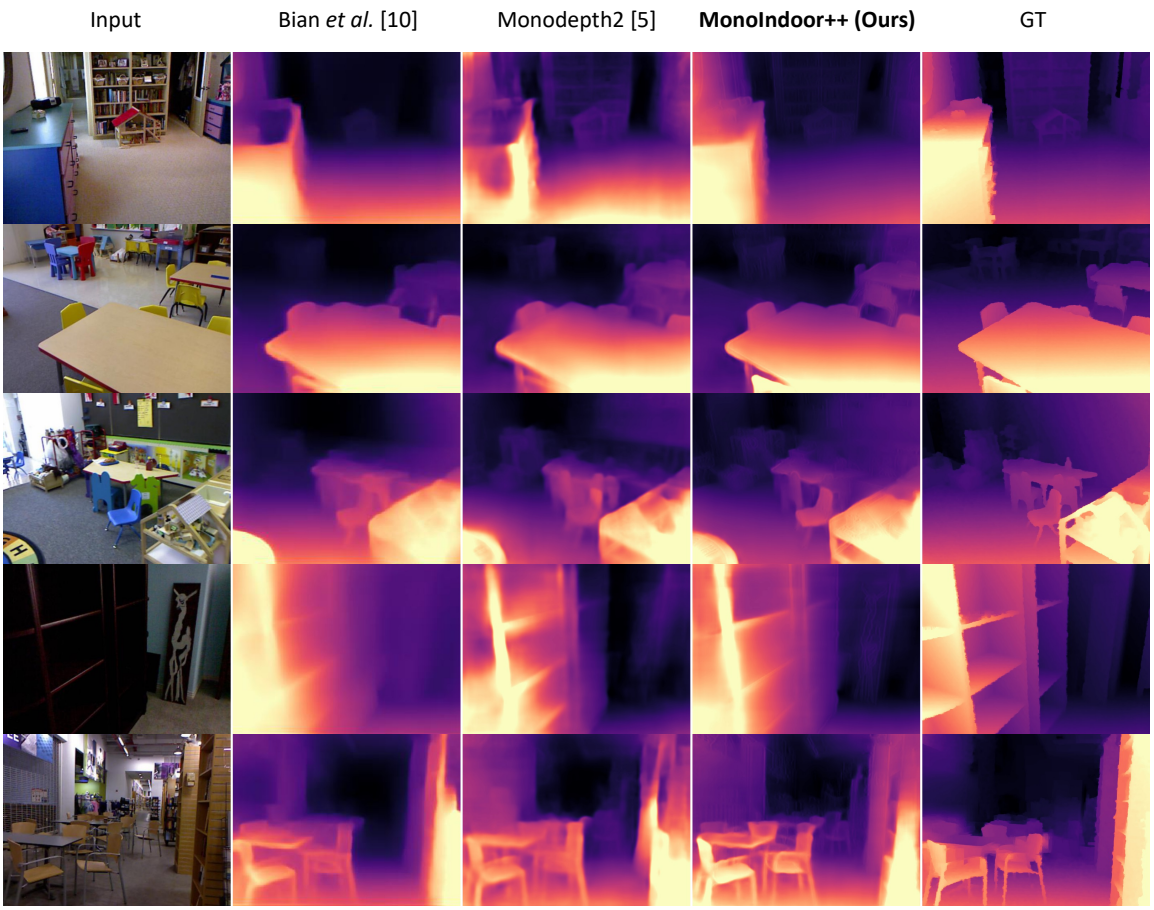

Figure 5.3: Qualitative comparison on NYUv2 [376]. Images form the left to the right are: input, depth from [31], [141], **MonoIndoor++(Ours)**, ground-truth depth. Compared with both the baseline method Monodepth2 [141] and recent work [31], our model produces accurate depth maps that are closer to the ground-truth.

Results on EuRoC MAV Dataset: In this sub-section, we present evaluation results of self-supervised monocular depth estimation on the EuRoC MAV dataset [42]. As there are not many

Table 5.2: Quantitative results and comparison between our **MonoIndoor++** with existing self-supervised methods on the test sequences V1_03, V2_01 V2_02, V2_03 of EuRoC MAV [42]. Best results are in **bold**.

Method	V1_03					V2_01				
	Error Metric ↓		Accuracy Metric ↑			Error Metric ↓		Accuracy Metric ↑		
	AbsRel	RMSE	δ_1	δ_2	δ_3	AbsRel	RMSE	δ_1	δ_2	δ_3
Bian *et al.* [29]	0.100	0.387	0.905	0.985	0.996	0.153	0.554	0.807	0.944	0.984
P^2Net [460]	0.104	0.387	0.905	0.986	0.997	0.155	0.557	0.780	0.953	0.989
Bian *et al.* [31]	0.094	0.360	0.925	0.985	0.995	0.148	0.536	0.800	0.950	0.987
Monodepth2 [141] (Baseline)	0.110	0.413	0.889	0.983	0.996	0.157	0.567	0.786	0.941	0.986
MonoIndoor [187] (Our ICCV21)	0.080	0.309	0.944	0.990	0.998	0.125	0.466	0.840	0.965	**0.993**
MonoIndoor++ (Ours)	**0.079**	**0.303**	**0.949**	**0.991**	**0.998**	**0.115**	**0.439**	**0.861**	**0.972**	0.992
Method	V2_02					V2_03				
	Error Metric ↓		Accuracy Metric ↑			Error Metric ↓		Accuracy Metric ↑		
	AbsRel	RMSE	δ_1	δ_2	δ_3	AbsRel	RMSE	δ_1	δ_2	δ_3
Bian *et al.* [29]	0.161	0.682	0.769	0.942	0.983	0.163	0.616	0.760	0.948	0.989
P^2Net [460]	0.150	0.604	0.800	0.955	0.989	0.152	0.541	0.792	0.954	0.991
Bian *et al.* [31]	0.154	0.637	0.783	0.948	0.987	0.149	0.534	0.792	0.962	0.992
Monodepth2 [141] (Baseline)	0.156	0.645	0.776	0.945	0.985	0.171	0.620	0.734	0.944	0.988
MonoIndoor [187] (Our ICCV21)	0.142	0.581	0.802	0.952	0.990	0.140	0.502	0.810	0.964	0.993
MonoIndoor++ (Ours)	**0.133**	**0.551**	**0.830**	**0.964**	**0.991**	**0.134**	**0.482**	**0.829**	**0.967**	**0.993**

public results on the EuRoC MAV dataset, excepting for comparing between our **MonoIndoor++** and the baseline method Monodepth2 [141], we follow official implementations of Bian *et al.* [29] [1], P^2Net [460] [2] and Bian *et al.* [31] [3] to conduct experiments and make fair comparisons.

Quantitative Results: We present quantitative results of our model **MonoIndoor++** and comparisons with other methods for the self-supervised monocular depth estimation on all Vicon room testing sequences in Table 5.2. It can be observed that, when compared with recent self-supervised methods [29, 460, 31], our model achieves the best performance across all major evaluation metrics (AbsRel and δ_1) on various scenes including the "difficult" scene, *i.e.*, "Vicon room 203" (V2_03). Specifically, on the sequence V2_01, our model improves self-supervised monocular depth estimation performance significantly by reducing the AbsRel to 11.5% and increasing the δ_1

[1] https://github.com/JiawangBian/SC-SfMLearner-Release
[2] https://github.com/svip-lab/Indoor-SfMLearner
[3] https://github.com/JiawangBian/sc_depth_pl

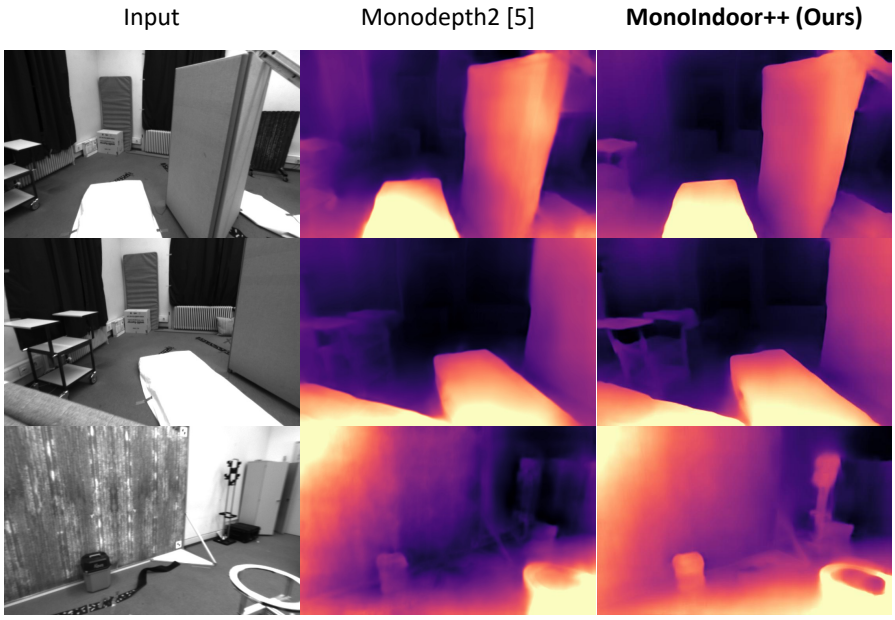

Figure 5.4: Qualitative comparison of depth prediction on EuRoC MAV. Our **MonoIndoor++** produces more accurate and cleaner depth maps.

to 86.1%. Similar improvements can be observed on other test sequences. Besides, compared with our preliminary work [187], our method consistently and significantly improves depth estimation performance across all test sequences. In addition, ablation studies for the effectiveness of each core module are presented in Section 5.4.5 and ablation experiments of the design choices for the scale network are shown in Section 5.4.5.

Qualitative Results: We present the qualitative results and comparisons of depth maps predicted by the baseline method Monodepth2 [141] and our **MonoIndoor++** in Figure 5.4. There are no ground-truth dense depth maps on the EuRoC MAV dataset. From Figure 5.4, it is clear that the depth maps generated by our model are much better than the ones by Monodepth2. For instance, in the third row, our model can predict precise depths for the ***hole*** region at the right-bottom corner

Table 5.3: Relative pose evalution on EuRoC MAV [364]. Results show the average absolute trajectory error(ATE), and the relative pose error(RPE) in meters and degrees, respectively. Scene: test sequence name.

Scene	Methods	ATE (m) ↓	RPE (m) ↓	RPE (°) ↓
V1_03	Monodepth2 [141] (Baseline)	0.0681	0.0686	1.3237
	MonoIndoor [187] (Our ICCV21)	0.0564	0.0638	0.7185
	MonoIndoor++ (Ours)	**0.0557**	**0.0542**	**0.5599**
V2_01	Monodepth2 [141] (Baseline)	0.0266	0.0199	1.1985
	MonoIndoor [187] (Our ICCV21)	0.0230	0.011	1.197
	MonoIndoor++ (Ours)	**0.0229**	**0.0050**	**1.1239**
V2_02	Monodepth2 [141] (Baseline)	0.0624	0.0481	**6.4135**
	MonoIndoor [187] (Our ICCV21)	0.0544	0.0360	7.2100
	MonoIndoor++ (Ours)	**0.0517**	**0.0350**	7.1928
V2_03	Monodepth2 [141] (Baseline)	0.0670	**0.0355**	5.3532
	MonoIndoor [187] (Our ICCV21)	0.0699	0.0748	5.304
	MonoIndoor++ (Ours)	**0.0644**	0.0676	**4.8559**

whereas such a hole structure in the depth map by Monodepth2 is missing. These observations are also consistent with the better quantitative results in Table 5.2, proving the superiority of our model.

Relative Pose Evaluation: In Table 5.3, we evaluate the proposed residual pose estimation module on all Vicon room test sequences V1_03, V2_01, V2_02 and V2_03 of the EuRoC MAV [42]. We follow [469] to evaluate relative camera poses estimated by our residual pose estimation module. We use the following evaluation metrics: absolute trajectory error (ATE) which measures the root-mean square error between predicted camera poses and ground-truth, and relative pose error (RPE) which measures frame-to-frame relative pose error in meters and degrees, respectively. As shown in Table 5.3, compared with the baseline model Monodepth2 [141] which employs one-stage pose network, using our method leads to improved relative pose estimation across evaluation metrics on most test scenes. Specifically, on the scene V1_03, the ATE by our MonoIndoor++ is significantly decreased from 0.0681 meters to **0.0557** meters and RPE(°) is reduced from 1.3237° to **0.5599**°. Similar observations are made on the scene V2_02, where the ATE by our MonoIndoor++

is significantly decreased from 0.0624 meters to **0.0517** meters. When compared with our preliminary work [187], consistent improvements can also be observed across almost all testing sequences, which can further validate the superiority of our model for self-supervised monocular depth estimation.

Table 5.4: Comparison of our method with existing supervised and self-supervised methods on ScanNet [91]. Best results among supervised and self-supervised methods are in **bold**.

Methods	Supervision	AbsRel ↓	SqRel ↓	RMS ↓	$\text{RMS}_{\log}$ ↓
Photometric BA [7]	✓	0.268	0.427	0.788	0.330
DeMoN [409]	✓	0.231	0.520	0.761	0.289
BANet [390]	✓	0.161	0.092	0.346	0.214
DeepV2D [394]	✓	0.069	**0.018**	0.196	0.099
NeuralRecon [386]	✓	**0.047**	0.024	**0.164**	**0.093**
Bian *et al.* [29]	✗	0.177	0.238	0.552	0.220
P^2Net [460]	✗	0.218	0.190	0.531	0.256
Bian *et al.* [31]	✗	0.163	0.096	0.428	0.188
Gu *et al.* [147]	✗	0.140	0.127	0.496	0.212
Monodepth2 [141] (Baseline)	✗	0.189	0.111	0.426	0.225
MonoIndoor [187] (Our ICCV21)	✗	0.126	0.057	0.329	0.163
MonoIndoor++ (Ours)	✗	**0.113**	**0.048**	**0.302**	**0.148**

Results on ScanNet Dataset: In this sub-section, we evaluate our **MonoIndoor++** and compare its performance with recent SOTA methods on the ScanNet dataset [91]. Referring to [390], the raw dataset is firstly downsampled 10 times along the temporal dimension and then $\sim 100K$ images are randomly selected for training. During testing, $\sim 4K$ are sampled from 100 different testing scenes to evaluate the trained model. It should be mentioned that we have observed that rarely research work have conducted thorough experiments on ScanNet for self-supervised monocular depth estimation. Instead, previous work [460, 31] simply conduct zero-shot generalization experiments. In this paper, we *first* present self-supervised depth estimation evaluation results, and *second*, we show evaluations of the zero-shot generalization on depth and relative pose

estimation. As introduced in Section 5.4.4, we follow official implementations of Bian *et al.* [29], P^{2}Net [460] and Bian *et al.* [31] to conduct experiments and make fair comparisons.

Self-supervised Depth Estimation Evaluation Table 5.4 presents the quantitative results of our model **MonoIndoor++** and both SOTA supervised and self-supervised methods on ScanNet. It shows that our **MonoIndoor++** outperforms the previous self-supervised methods [141, 460, 31, 147] in depth estimation, reaching the best results across all metrics. For instance, our model gives 11.3% of the AbsRel, which is exceptionally competitive in indoor environments. When compared with our preliminary work [187], our MonoIndoor++ consistently improves depth estimation performance on this challenging dataset. In addition to that, our model outperforms a group of supervised methods [409, 390]. Ablation studies for the effectiveness of each core module are presented in Section 5.4.5.

Table 5.5: Zero-shot generalization of our method for self-supervised depth estimation on ScanNet [91]. Best results are in **bold**.

Methods	Supervision	Error Metric ↓		Accuracy Metric ↑		
		AbsRel	RMS	δ_1	δ_2	δ_3
Latina *et al.* [227]	✓	0.141	0.339	0.811	.958	0.990
VNL [455]	✓	**0.123**	**0.306**	**0.848**	**0.964**	**0.991**
Zhou *et al.* [495]	✗	0.212	0.483	0.650	0.905	0.976
Zhao *et al.* [483]	✗	0.179	0.415	0.726	0.927	0.980
Bian *et al.* [29]	✗	0.169	0.392	0.749	0.938	0.983
P^{2}Net [460]	✗	0.175	0.420	0.740	0.932	0.982
Bian *et al.* [31]	✗	0.156	0.361	0.781	0.947	0.987
Monodepth2 [141] (Baseline)	✗	0.170	0.401	0.730	0.948	0.991
MonoIndoor [187] (Our ICCV21)	✗	0.154	0.373	0.779	0.951	0.988
MonoIndoor++ (Ours)	✗	**0.138**	**0.347**	**0.810**	**0.967**	**0.993**

Zero-shot Generalization We present the zero-shot generalization results of self-supervised depth estimation on ScanNet [91] in Table 5.5, where we evaluate the proposed **MonoIndoor++** pretrained on NYUv2 dataset. From Table 5.5, it is observed that our NYUv2 pretrained model generalizes better than other recent methods to new dataset. Besides, we show the zero-shot gener-

alization results of relative pose estimation on ScanNet in Table 5.6. We follow [394, 460, 31] to use 2000 image pairs selected from diverse indoor scenes for pose evaluation. It can be observed that our method outperforms other self-supervised methods. Specifically, compared to Bian *et al.* [31], our method significantly reduces translational error (tr (cm)) from 0.55 centimeters to **0.27** centimeters and decreases camera rotational error (rot (deg)) from 1.82 to **1.19**. When compared with our preliminary work [187], consistent improvements are observed on depth and relative pose evaluation results. Both depth and pose results validate the good zero-shot generalizability and capability of our method.

Table 5.6: Zero-shot generalization of our method for relative pose estimation on ScanNet [91]. Best results are in **bold**. rot:rotational error of the relative pose. tr: translational error of the relative pose.

Methods	rot (deg) ↓	tr (deg) ↓	tr (cm) ↓
Zhou *et al.* [495]	1.96	39.17	1.4
P^2Net [460]	1.86	35.11	0.89
Bian *et al.* [31]	1.82	39.41	0.55
Monodepth2 [141] (Baseline)	2.03	41.12	0.83
MonoIndoor [187] (Our ICCV21)	1.36	23.42	1.04
MonoIndoor++ (Ours)	**1.19**	**21.33**	**0.27**

Results on RGB-D 7-Scenes Dataset In this sub-section, we evaluate our **MonoIndoor++** on the RGB-D 7-Scenes dataset [373] under two settings, the zero-shot generalization and the fine-tuning strategy, respectively. Following [30, 31], we extract one image from every 30 frames in each video sequence. For fine-tuning, we first pre-train our model on the NYUv2 dataset, and then fine-tune the pre-trained model on each scene of 7-Scenes dataset.

Table 5.7: Comparison of our method to latest self-supervised methods under zero-shot generalization and fine-tuning settings on RGB-D 7-Scenes [373]. Best results are in **bold**.

Scenes	Zero-shot Generalization							
	Bian *et al.* [31]		Bian *et al.* [30]		Monodepth2 [141]		**MonoIndoor++ (Ours)**	
	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑
Chess	0.179	0.689	0.169	0.719	0.193	0.654	**0.157**	**0.750**
Fire	0.163	0.751	0.158	0.758	0.190	0.670	**0.150**	**0.768**
Heads	0.171	0.746	**0.162**	**0.749**	0.206	0.661	0.171	0.727
Office	0.146	0.799	0.132	0.833	0.168	0.748	**0.130**	**0.837**
Pumpkin	0.120	0.841	0.117	0.857	0.135	0.816	**0.102**	**0.895**
RedKitchen	0.136	0.822	0.151	0.780	0.168	0.733	**0.144**	**0.795**
Stairs	**0.143**	0.794	0.162	0.765	0.146	**0.806**	0.155	0.753
Scenes	After Fine-tuning							
	Bian *et al.* [31]		Bian *et al.* [30]		Monodepth2 [141]		**MonoIndoor++ (Ours)**	
	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑	AbsRel ↓	Acc (δ_1) ↑
Chess	0.150	0.780	0.103	0.880	0.123	0.853	**0.097**	**0.888**
Fire	0.105	0.918	0.089	0.916	0.091	0.927	**0.077**	**0.939**
Heads	0.143	0.833	0.124	0.862	0.130	0.855	**0.106**	**0.889**
Office	0.128	0.855	0.096	0.912	0.105	0.897	**0.083**	**0.934**
Pumpkin	0.097	0.922	0.083	**0.946**	0.116	0.877	**0.078**	0.945
RedKitchen	0.124	0.853	0.101	0.896	0.108	0.884	**0.094**	**0.915**
Stairs	0.134	0.823	0.106	0.855	0.127	0.825	**0.104**	**0.857**

Table 5.7 presents the quantitative results and comparisons of our model **MonoIndoor++** and latest SOTA self-supervised methods on 7-Scenes dataset. It can be observed that our model outperforms the baseline method Monodepth2 [141] significantly on each scene. Further, compared to the model [30, 31], our method achieve the best performance on most scenes before and after fine-tuning using NYUv2 pretrained models, which demonstrates better generalizability and capability of our model. Moreover, the results show that our method can perform well in a variety of different scenes.

5.4.5 Ablation Studies

Effects of each proposed module in MonoIndoor++: In this sub-section, we perform ablation studies of each core module in our proposed **MonoIndoor++** on NYUv2 [376], ScanNet [91] and EuRoC MAV [42] datasets in Table 5.8.

Specifically, We first perform ablation study for the residual pose estimation module. In Table 5.8, from methods of the "Monodepth2 [141] (Baseline)" and "**MonoIndoor++ (Ours)**" with the "Residual Pose" column checked, improved performance can be observed by using the proposed residual pose estimation module. For instance, on NYUv2, the AbsRel is decreased from 16% to 14.2% and δ_1 is increased from 76.7% to 81.3%; on ScanNet, the AbsRel is decreased from 18.9% to 13.6% and the δ_1 is increased from 70.9% to 83.3%; on EuRoC MAV V2_01, the AbsRel is decreased from 15.7% to 14.1% and the δ_1 is increased from 78.6% to 81.5% and similar observations can be made on other test sequences as well.

Next, we experiment to validate the effectiveness of the depth factorization module. Comparing with Monodepth2 which predicts depth without any guidance of global scales, by adding the depth factorization module with a separate scale network in our MonoIndoor++ (see "**MonoIndoor++ (Ours)**" with the "Residual Pose" and "Depth Factorization" columns checked), we further observe improved performance on all datasets. For instance, on NYUv2, the AbsRel is decreased from 14.2% to 13.4% and δ_1 is increased from 81.3% to 82.3%; on ScanNet, the AbsRel is decreased from 13.6% to 12.6% and the δ_1 is increased from 83.3% to 83.9%; on EuRoC MAV V2_01, the AbsRel is decreased from 14.1% to 12.5% and the δ_1 is increased from 81.5% to 84.0% and similar observations can be made on other test sequences.

In addition, by using the residual pose estimation module with both the proposed depth factorization module and coordinates convolutional encoding module (see "**MonoIndoor++ (Ours)**" with all columns checked, the performance can be improved consistently. For instance, on NYUv2, the AbsRel is decreased to 13.2% and δ_1 is increased to 83.4%; on ScanNet, the AbsRel is decreased to 11.3% and the δ_1 is increased to 87.3%; on EuRoC MAV V2_01, the AbsRel is decreased

to 11.5% and the δ_1 is increased to 86.1% and similar observations can be made on other test sequences as well. Our full model achieves the best performance by giving competitive depth esitmation results on these challenging datasets. We argue that these ablation results clearly prove the effectiveness of the each proposed module in our model, **MonoIndoor++**.

Table 5.8: Ablation results on each core module of our **MonoIndoor++** and comparison with the baseline method on the NYUv2 [376], ScanNet [91] and EuRoC MAV [42] datasets. Best results are in **bold**. Residual Pose: our residual pose estimation module. Depth Factorization: our depth factorization module with scale network. Coordinates Conv. Encoding: our coordinates convolutional encoding module.

Method	Residual Pose	Depth Factorization	Coordinates Conv. Encoding	NYUv2					ScanNet				
				Error Metric ↓		Accuracy Metric ↑			Error Metric ↓		Accuracy Metric ↑		
				AbsRel	RMSE	δ_1	δ_2	δ_3	AbsRel	RMSE	δ_1	δ_2	δ_3
Monodepth2 [141] (Baseline)	✗	✗	✗	0.16	0.601	0.767	0.949	0.988	0.189	0.426	0.709	0.929	0.984
MonoIndoor++ (Ours)	✓	✗	✗	0.142	0.553	0.813	0.958	0.988	0.136	0.345	0.833	0.968	0.995
MonoIndoor++ (Ours)	✓	✓	✗	0.134	0.526	0.823	0.958	0.989	0.126	0.329	0.839	0.973	0.995
MonoIndoor++ (Ours)	✓	✓	✓	**0.132**	**0.517**	**0.834**	**0.961**	**0.990**	**0.113**	**0.302**	**0.873**	**0.979**	**0.996**

Method	Residual Pose	Depth Factorization	Coordinates Conv. Encoding	EuRoC MAV V1_03					EuRoC MAV V2_01				
				Error Metric ↓		Accuracy Metric ↑			Error Metric ↓		Accuracy Metric ↑		
				AbsRel	RMSE	δ_1	δ_2	δ_3	AbsRel	RMSE	δ_1	δ_2	δ_3
Monodepth2 [141] (Baseline)	✗	✗	✗	0.110	0.413	0.889	0.983	0.996	0.157	0.567	0.786	0.941	0.986
MonoIndoor++ (Ours)	✓	✗	✗	0.100	0.379	0.913	0.987	0.997	0.141	0.518	0.815	0.961	0.991
MonoIndoor++ (Ours)	✓	✓	✗	0.080	0.309	0.944	0.990	0.998	0.125	0.466	0.840	0.965	**0.993**
MonoIndoor++ (Ours)	✓	✓	✓	**0.079**	**0.303**	**0.949**	**0.991**	**0.998**	**0.115**	**0.439**	**0.861**	**0.972**	0.992

Method	Residual Pose	Depth Factorization	Coordinates Conv. Encoding	EuRoC MAV V2_02					EuRoC MAV V2_03				
				Error Metric ↓		Accuracy Metric ↑			Error Metric ↓		Accuracy Metric ↑		
				AbsRel	RMSE	δ_1	δ_2	δ_3	AbsRel	RMSE	δ_1	δ_2	δ_3
Monodepth2 [141] (Baseline)	✗	✗	✗	0.156	0.645	0.776	0.945	0.985	0.171	0.620	0.734	0.944	0.988
MonoIndoor++ (Ours)	✓	✗	✗	0.150	0.619	0.792	0.950	0.988	0.147	0.538	0.806	0.963	0.989
MonoIndoor++ (Ours)	✓	✓	✗	0.142	0.581	0.802	0.952	0.990	0.140	0.502	0.810	0.964	0.993
MonoIndoor++ (Ours)	✓	✓	✓	**0.133**	**0.551**	**0.830**	**0.964**	**0.991**	**0.134**	**0.482**	**0.829**	**0.967**	**0.993**

We also present the exemplar depth visualizations by our proposed modules on NYUv2 dataset in Figure 5.5. In addition, we visualize intermediate and final synthesized views compared with the current view on NYUv2 in the Figure 5.6. Highlighted regions show that final synthesized views are better than the intermediate synthesized views and closer to the current view.

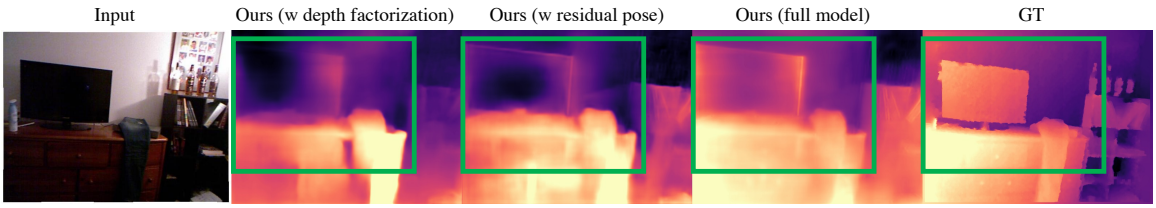

Figure 5.5: Qualitative ablation comparisons of depth prediction on NYUv2. Our full model with both depth factorization and residual pose modules produce better depth maps.

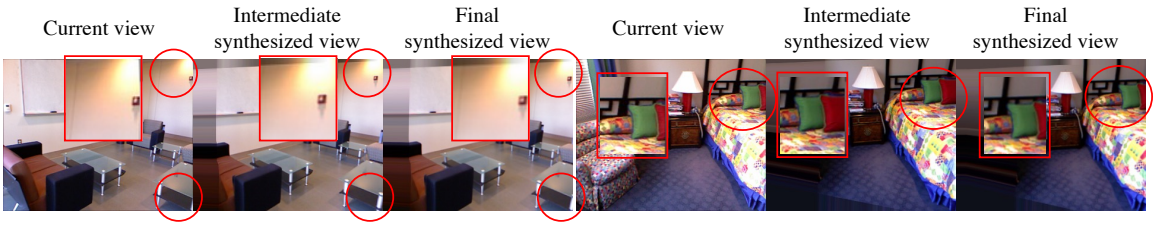

Figure 5.6: Intermediate synthesized views on NYUv2.

Effects of network design for transformer-based scale regression network: We perform ablation studies for our network design choices for the transformer-based scale regression network in depth factorization module on the test sequence V2_01 of the EuRoC MAV dataset [42]. Firstly, we consider the following designs as the backbone of our scale regression network: I) a pre-trained ResNet-18 [162] followed by a group of Convolutional-BN-ReLU layers; II) a pre-trained ResNet-18 [162] followed by two residual blocks; III) a lightweight network with two residual blocks which shares the feature maps from the depth encoder as input. These three choices are referred to as the ScaleCNN, ScaleNet and ScaleRegressor, respectively in Table 5.9. Next, we validate the effectiveness of adding new components into our backbone design. As described in Section 5.3.2, we mainly integrate two sub-modules: i) a transformer module and ii) a probabilistic scale regression block.

Table 5.9: Ablation results of design choices and the effectiveness of components in the transformer-based scale regression network of our model (**MonoIndoor++**) on EuRoC MAV V2_01 [42]. Porb. Reg.: the probabilistic scale regression block. Note: we only use the residual pose estimation module when experimenting with different network designs for the depth factorization module.

Network Design	Attention	Prob. Reg.	Error Metric ↓		Accuracy Metric ↑		
			AbsRel	RMSE	δ_1	δ_2	δ_3
I. ScaleCNN	✓	✓	0.140	0.518	0.821	0.956	0.985
II. ScaleNet	✓	✓	0.141	0.519	0.817	0.959	0.988
III. ScaleRegressor	✗	✗	0.139	0.508	0.817	0.960	0.987
III. ScaleRegressor	✓	✗	0.135	0.501	0.825	0.964	0.989
III. ScaleRegressor	✓	✓	**0.125**	**0.466**	**0.840**	**0.965**	**0.993**

As shown in Table 5.9, the best performance is achieved by ScaleRegressor that uses transformer module and probabilistic scale regression. It proves that sharing features with the depth encoder is beneficial to scale estimation. Comparing the results of three ScaleRegressor variants, the performance gradually improves as we add more components (*i.e.*., attention and probabilistic scale regression (Prob. Reg.)). Specifically, adding the transformer module improves the overall performance over the baseline backbone; adding the probabilistic regression block leads to a further improvement, which validates the effectiveness of our proposed sub-modules.

Ablation results of coordinates convolutional encoding: We present ablation studies for the encoding position of the coordinates convolutional encoding module on the NYUv2 [376] and V2_01 of the EuRoC MAV [42] datasets in Tabel 5.10. It should be mentioned that, to fully explore the effectiveness of using coordinates encoding technique, we only run our **MonoIndoor++** with the residual pose estimation module. We perform coordinates convolutional encoding with the following choices. Specifically, we first encode coordinates information with the color image pairs and extend coordinates convolutional layers to process combined input data. Second, we perform coordinates encoding operations with the feature representations outputted from the pose

encoder and the processed features are taken as the input to the pose decoder. Third, we incorporate coordinates encoding operations with both input and features from pose encoder for pose estimation.

Table 5.10: Ablation results of encoding position for coordinates conovlutional with our **MonoIndoor++** on NYUv2. Init.: initialization of weights. Note: we only use the residual pose estimation module when experimenting with different network designs for the coordinates convolutional encoding module.

Model	Encoding Position	NYUv2				
		Error Metric ↓		Accuracy Metric ↑		
		AbsRel	RMS	δ_1	δ_2	δ_3
MonoIndoor	✗	0.142	0.553	0.813	0.958	0.988
MonoIndoor++ (Random Init.)	Input	0.140	**0.543**	0.817	0.959	0.989
MonoIndoor++ (ImageNet Init.)	Input	**0.139**	0.545	**0.821**	**0.958**	**0.989**
MonoIndoor++ (ImageNet Init.)	Encoder Features	0.145	0.565	0.806	0.954	0.988
MonoIndoor++ (ImageNet Init.)	Input & Encoder Features	0.141	0.554	0.815	0.957	0.989
Model	Encoding Position	EuRoC MAV V2_01				
		Error Metric ↓		Accuracy Metric ↑		
		AbsRel	RMS	δ_1	δ_2	δ_3
MonoIndoor	✗	0.141	0.518	0.786	0.941	0.986
MonoIndoor++	Input	**0.130**	**0.492**	**0.840**	**0.965**	**0.992**

From the Table 5.10, it can be observed that, by using coordinates convolutional encoding in residual pose estimation module, performance can be improved. For instance, the AbsRel is decreased to **13.9%** from 14.2% and the δ_1 is improved from 81.7% to **82.1%**. Besides, comparing with encoding coordinates information with feature representations after the pose encoder, applying the coordinates convolutional encoding operation over the input image pairs directly gives the best performance. Further, we test two different initialization methods for coordinates convolutional layers which are with random initializations or ImageNet-pretrained [95] initialization, respectively. The coordinates convolutional encoding layers which are initialized with ImageNet-pretrained weights give slightly improved performance compared to ones with random weights.

Given the above observations, we further perform experiments under the same settings on EuRoC MAV V2_01 sequence, significant improvements have been observed for self-supervised monocular depth estimation by using our residual pose estimation module with coordinates convolutional encoding module, which can further validate the effectiveness of the coordinates convolutional encoding module.

Chapter 6

Learning Local Recurrent Models for Human Mesh Recovery

6.1 Introduction

We consider the problem of human mesh recovery in videos, i.e., fitting a parametric 3D human mesh model to each frame of the video. With many practical applications [381, 291], including in healthcare for COVID-19 [235, 75, 196], there has been much progress in this field in the last few years [193, 215, 133]. In particular, most research effort has been expended in single image-based mesh estimation where one seeks to fit the human mesh model to a single image. However, such 3D model estimation from only a single 2D projection (image) is a severely under-constrained problem since multiple 3D configurations (in this case poses and shapes of the mesh model) can project to the same image. Such ambiguities can be addressed by utilizing an extra dimension that

is typically associated with images- the temporal dimension leading to video data and the problem of video mesh recovery.

Figure 6.1: We present **LMR**, a new method for video human mesh recovery. Unlike existing work, LMR captures local human part dynamics and interdependencies by learning multiple local recurrent models, resulting in notable performance improvement over the state of the art. Here, we show a few qualitative results on the 3DPW dataset.

The currently dominant paradigm for video mesh recovery involves the *feature-temporal-regressor* architecture. A deep convolutional neural network (CNN) is used to extract frame-level image feature vectors, which are then processed by a temporal encoder to learn the motion dynamics in the video. The representation from the temporal encoder is then processed by a parameter regressor module that outputs frame-level mesh parameter vectors. While methods vary in the specific implementation details, they mostly follow this pipeline. For instance, while Kanazawa *et al.* [194] implement the temporal encoder using a feed-forward fully convolutional model, Kocabas *et al.* [215] uses a recurrent model to encode motion dynamics. However, uniformly across all these methods, the parameter regressor is implemented using a "flat" regression architecture that takes in feature vectors as input and directly regresses all the model parameters, e.g., 85 values (pose, shape, and camera) for the popularly used skinned multi-person linear (SMPL) model [280, 193]. While this paradigm has produced impressive recent results as evidenced by the mean per-joint position

errors on standard datasets (see Arnab *et al.* [13] and Kocabas *et al.* [215] for a fairly recent benchmark), a number of issues remain unaddressed that provide us with direction and scope for further research and performance improvement.

First, the above architectures implicitly assume that all motion dynamics can be captured using a single dynamical system (e.g., a recurrent network). While this assumption may be reasonable for fairly simplistic human motions, it is not sufficient for more complex actions. For instance, while dancing, the motion dynamics of a person vary from one part of the body to the other. As a concrete example, the legs may remain static while the hands move vigorously, and these roles may be reversed after a certain period of time (static hands and moving legs several frames later), leading to more "locally" varying dynamics. Intuitively, this tells us that the motion of each local body part should in itself be modeled separately by a dynamical system, and that such a design should help capture this local "part-level" dynamical information more precisely as opposed to a single dynamical system for the entire video snippet.

Next, as noted above, the *regressor* in the *feature-temporal-regressor* architecture involves computing all the parameters of the SMPL model using a direct/flat regression design without due consideration given to the interdependent nature of these parameters (i.e., SMPL joint rotations are not independent but rather conditioned on other joints of other parts such as the root [280]). It has been noted in prior work [204] that such direct regression of rotation matrices, which form a predominant part of the SMPL parameter set, is challenging as is and only made further difficult due to these interdependencies in the SMPL model. In addition to direct rotation regression, the temporal module in the above *feature-temporal-regressor* also does not consider any joint and part

interdependencies, i.e., modeling all motion dynamics using a single global dynamical system, thus only further exacerbating this problem.

To address the aforementioned issues, we present a new architecture for capturing the human motion dynamics for estimating a parametric mesh model in videos. Please note that while we use the SMPL model [280] in this work, our method can be extensible to other kinds of hierarchical parametric human meshes as well. See Figure 6.1 for some qualitative results with our method on the 3DPW [416] dataset and Figure 6.2 for a comparison with a current state-of-the-art method. Our method, called *local recurrent models for mesh recovery* (**LMR**), comprises several design considerations. First, to capture the need for modeling locally varying dynamics as noted above, LMR defines six local recurrent models (root, head, left/right arms, left/right legs), one each to capture the dynamics of each part. As we will describe later, each "part" here refers to a chain of several joints defined on the SMPL model. Note that such a part division is not ad hoc but grounded in the hierarchical and part-based design of the SMPL model itself, which divides the human body into the six parts above following the standard skeletal rigging procedure [280]. Next, to model the conditional interdependence of local part dynamics, LMR first infers root part dynamics (i.e., parameters of all joints in the root part). LMR then uses these root part parameters to subsequently infer the parameters of all other parts, with the output of each part conditioned on the root output. For instance, the recurrent model responsible for producing the parameters of the left leg takes as input both frame-level feature vectors as well as frame-level root-part parameters from the root-part recurrent model.

Note the substantial differences between LMR's design and those of prior work- (a) we use multiple local recurrent models instead of one global recurrent model to capture motion dynam-

ics, and (b) such local recurrent modeling enables LMR to explicitly capture local part dependencies. Modeling these local dependencies enables LMR to infer motion dynamics and frame-level video meshes informed by the geometry of the problem, i.e., the SMPL model, which, as noted in prior work [204], is an important design consideration as we take a step towards accurate rotation parameter regression architectures. We conduct extensive experiments on a number of standard video mesh recovery benchmark datasets (Human3.6M [182], MPI-INF-3DHP [295], and 3DPW [416]), demonstrating the efficacy of such local dynamic modeling as well as establishing state-of-the-art performance with respect to standard evaluation metrics.

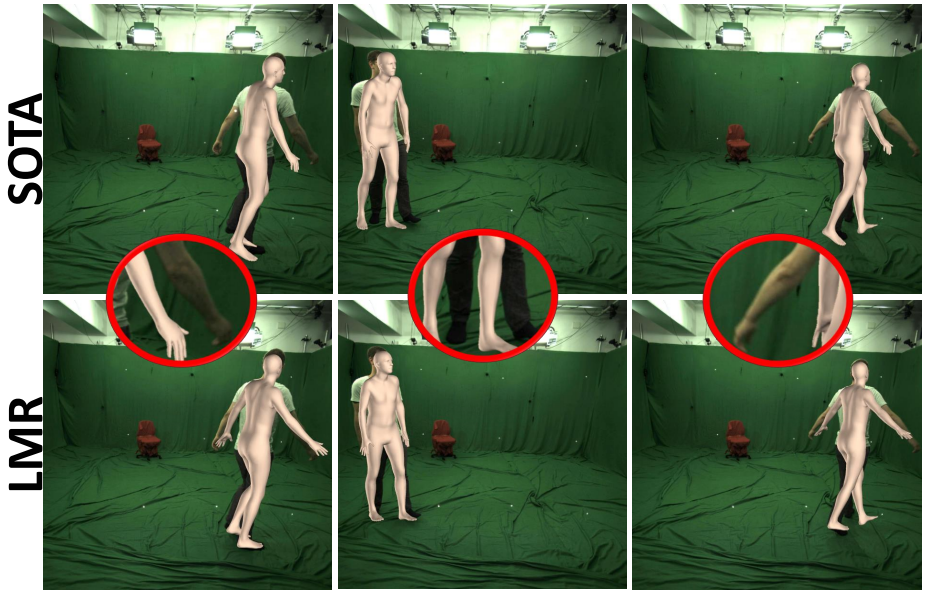

Figure 6.2: A qualitative comparison with VIBE [215], highlighting local regions (ellipses that show zoomed-in VIBE results) where LMR gives better performance.

6.2 Related Work and Our Contributions

There is much recent work in human pose estimation, including estimating 2D keypoints [303, 47, 470], 3D keypoints [290, 316, 154, 183, 434, 435, 433], and a full mesh [193, 317, 194, 217, 13, 133, 215]. Here, we discuss methods that are relevant to our specific problem- fitting 3D meshes to image and video data.

6.2.1 Single-image mesh fitting

Most recent progress in human mesh estimation has been in fitting parametric meshes to single image inputs. In particular, following the availability of differentiable parametric models such as SMPL [280], there has been an explosion in interest and activity in this field. Kanazawa *et al.* [193] presented an end-to-end trainable regression architecture for this problem that could in principle be trained with 2D-only keypoint data. Subsequently, many improved models have been proposed. Kolotourous *et al.* [217] and Georgakis *et al.* [133] extended this architecture to include more SMPL-structure-informed design considerations using either graph-based or parameter factorization-based approaches. Sun *et al.* [387] and Lin *et al.* [253] studied one-stage human mesh recovery using either transformer-based model or center heatmap to help to describe 3D body mesh. There have also been attempts at SMPL-agnostic modeling of joint interdependencies, with Fang *et al.* [120] employing bidirectional recurrent networks and Isack *et al.* [184] learning priors between joints using a pre-defined joint connectivity scheme. While methods such as Georgakis *et al.* [133] and Zhou *et al.* [499] also take a local part-based kinematic approach, their focus is on capturing inter-joint spatial dependencies. On the other hand, LMR's focus is on capturing inter-part temporal dependencies which LMR models using separate recurrent networks.

6.2.2 Video mesh fitting

Following the success of image-based mesh fitting methods, there has been a recent uptick in interest and published work in fitting human meshes to videos. Arnab *et al.* [13] presented a two-step approach that involved generating 2D keypoints and initial mesh fits using existing methods, and then using these initial estimates to further refine the results using temporal consistency constraints, e.g., temporal smoothness and 3D priors. However, such a two-step approach is susceptible to errors in either steps and our proposed LMR overcomes this issue with an end-to-end trainable method that provides deeper integration of the temporal data dimension both in training and inference. On the other hand, Kanazawa *et al.* [194] and Kocabas *et al.* [215] also presented end-to-end variants of the *feature-temporal-regressor* where frame-level feature vectors are first encoded using a temporal encoder (e.g., a single recurrent network) and finally processed by a parameter regressor to generate meshes. However, such a global approach to modeling motion dynamics (with only one RNN) does not capture the disparities in locally varying dynamics (e.g., hands vs. legs) which is typically the case in natural human motion. LMR addresses this issue by design with multiple local RNNs in its architecture, one for each pre-defined part of the human body. Such a design also makes mesh parameter regression more amenable by grounding this task in the geometry of the problem, i.e., the SMPL model itself.

6.2.3 Contributions of this Chapter

- We present LMR, the first local-dynamical-modeling approach to video mesh recovery where unlike prior work, we explicitly model the local dynamics of each body part with separate recurrent networks.

- Unlike prior work that regresses mesh parameters in a direct or "flat" fashion, our local recurrent design enables LMR to explicitly consider human mesh interdependencies in parameter inference, thereby resulting in a structure-informed local recurrent architecture.

- We conduct extensive experiments on standard benchmark datasets and report competitive performance, establishing state-of-the-art results in many cases.

6.2.4 Parametric Mesh Representation

We use the Skinned Multi-Person Linear (SMPL) model [280] to parameterize the human body. SMPL uses two sets of parameter vectors to capture variations in the human body: shape and pose. The shape of the human body is represented using a 10-dimensional vector $\boldsymbol{\beta} \in \mathbb{R}^{10}$ whereas the pose of the body is represented using a 72-dimensional vector $\boldsymbol{\theta} \in \mathbb{R}^{72}$. While $\boldsymbol{\beta}$ corresponds to the first ten dimensions of the PCA projection of a shape space, $\boldsymbol{\theta}$ captures, in axis-angle format [43], the global rotation of the root joint (3 values) and relative (to the root) rotations of 23 other body joints (69 values). Given $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and a learned model parameter set $\boldsymbol{\psi}$, SMPL defines the mapping $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) : \mathbb{R}^{82} \rightarrow \mathbb{R}^{3\times N}$ from the 82-dimensional parametric space to a vertex space of $N = 6890$ 3D mesh vertices. One can then infer the 24 3D joints of interest (e.g., hips, legs, etc.) $\boldsymbol{X} \in \mathbb{R}^{3\times K}, K = 24$ using a pre-learned joint regression matrix $\boldsymbol{W}$ as $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{J}$. Using a known

camera model, e.g., a weak-perspective model as in prior work [193], one can then obtain the corresponding 24 2D image points $\boldsymbol{x} \in \mathbb{R}^{2\times K}$ as:

$$\boldsymbol{x} = s\Pi(\boldsymbol{X}(\boldsymbol{\beta}, \boldsymbol{\theta})) + \boldsymbol{t}, \tag{6.1}$$

where the scale $s \in \mathbb{R}$ and translation $\boldsymbol{t} \in \mathbb{R}^2$ represent the camera model, and Π is an orthographic projection. Therefore, fitting 3D SMPL mesh to a single image involves estimating the parameter set $\boldsymbol{\Theta} = \{\boldsymbol{\beta}, \boldsymbol{\theta}, s, \boldsymbol{t}\}$. In video mesh recovery, we take this a step forward by estimating $\boldsymbol{\Theta}$ for every frame in the video.

6.2.5 Learning Local Recurrent Models

As noted in Section 7.1, existing video mesh fitting methods formulate the problem in the *feature-temporal-regressor* design where all motion dynamics in the video are captured using a single RNN. We argue that this is insufficient for mesh estimation due to the inherently complex nature of human actions/motion, more so in challenging in-the-wild scenarios. Our key insight is that natural human motion dynamics has a more locally varying characteristic that can more precisely be captured using locally learned recurrent networks. We then translate this idea into a conditional local recurrent architecture, called **LMR** and visually summarized in Figure 6.3, where we define multiple recurrent models, one each to capture the dynamics of the corresponding local region in the human body. During training and inference, LMR takes as input a segment of an input video $\boldsymbol{V} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_t, t = 1, 2, \ldots, T\}$, where T is a design parameter corresponding to the length of the input sequence. LMR first processes each frame with its feature extraction module to produce frame-level feature vectors $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_t\}$ for each of the T frames. LMR then processes $\boldsymbol{\Phi}$ with its local part-level recurrent models and associated parameter regressors, and

aggregates all part-level outputs to obtain the mesh and camera parameters $\mathbf{\Theta}_t, t = 1, 2, \ldots, T$ for each frame, finally producing the output video mesh.

LMR Architecture

As shown in Figure 6.3(a), our architecture comprises a feature extractor followed by our proposed LMR module. The LMR module is responsible for processing the frame-level representation $\mathbf{\Phi}$ to output the per-frame parameter vectors $\mathbf{\Theta}_t$. Following the design of the SMPL model and prior work [280, 133], we divide the human body into six local parts- *root* (4 joints in the root region), *head* (2 joints in the head region), *left arm* (5 joints on left arm), *right arm* (5 joints on right arm), *left leg* (4 joints on left leg), and *right leg* (4 joints on right leg). Given this division, the pose of local part $p_i, i = 1, \ldots, 6$ can be expressed as $\boldsymbol{\theta}^i = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{n_i}], i = 1, \ldots, 6$, where $\boldsymbol{r}_q$ ($q = 1, \ldots, n_i$) is a rotation parameterization (e.g., $\boldsymbol{r}_q \in \mathbb{R}^3$ in case of axis angle) of joint q and n_i is the number of joints defined in part i. The overall pose parameter vector $\boldsymbol{\theta}$ can then be aggregated as $\boldsymbol{\theta} = [\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^6]$.

6.3 Technical Approach

To capture locally varying dynamics across the video sequence, LMR defines one recurrent model for each of the six parts defined above (see Figure 6.3(b)). The recurrent model for part i is responsible for predicting its corresponding $\boldsymbol{\theta}^i$. To capture the conditional dependence between parts, the information propagation during training and inference is defined as follows. Given the frame-level feature representation $\mathbf{\Phi}$, the mean pose vector $\boldsymbol{\theta}_{\text{mean}}$, and the mean shape vector $\boldsymbol{\beta}_{\text{mean}}$ (note that it is common [193, 194, 215] to initialize mesh fitting with these mean values),

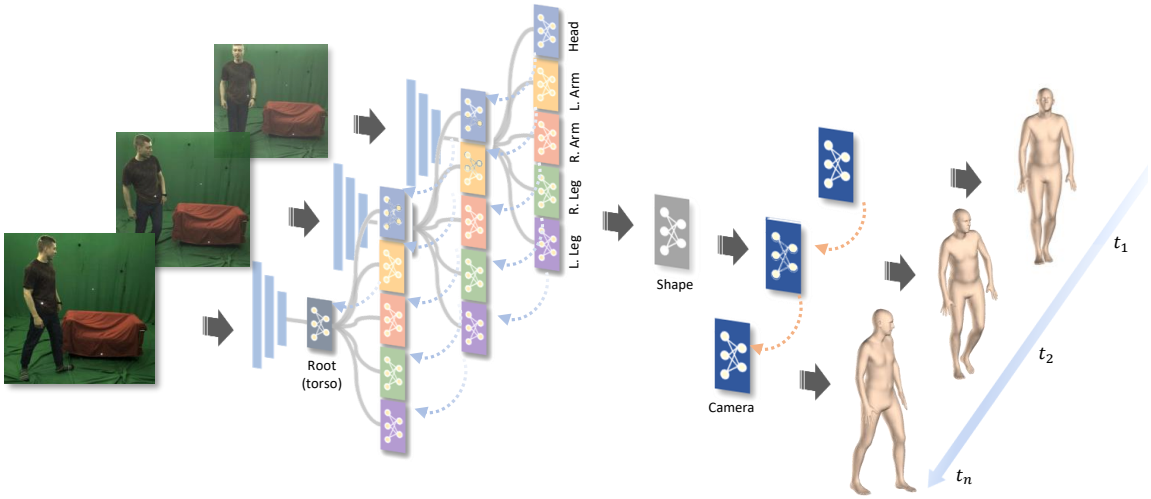

Figure 6.3: The proposed local recurrent modeling approach to human mesh recovery.

the recurrent model responsible for the root part (number 1) first predicts its corresponding pose vector $\boldsymbol{\theta}_t^1, t = 1, \ldots, T$ for each of the t frames using the concatenated vector $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_{\text{mean}}^1, \boldsymbol{\beta}_{\text{mean}}]$ as input for the current frame t. Note that $\boldsymbol{\Phi}_t$ is the feature vector for frame t and $\boldsymbol{\theta}_{\text{mean}}^1$ represents the mean pose parameters of part p_1. All other recurrent models (parts 2 through 6) then take in as input the concatenated vector $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_{\text{mean}}^k, \boldsymbol{\beta}_{\text{mean}}, \boldsymbol{\theta}_t^1]$ in predicting their corresponding pose vectors $\boldsymbol{\theta}_t^k, k = 2, \ldots, 6$ and $t = 1, \ldots, T$, where $\boldsymbol{\theta}_{\text{mean}}^k$ represents the mean pose parameters of part p_k. Note this explicit dependence of part k on the root (part 1) prediction $\boldsymbol{\theta}^1$. Given the aggregated (over all 6 parts) pose vector $\boldsymbol{\theta}_t$, LMR has a fully-connected module that takes as input the concatenated vector $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_t, \boldsymbol{\beta}_{\text{mean}}]$ for each frame t to predict the per-frame shape vectors $\boldsymbol{\beta}_t, t = 1, \ldots, T$. Finally, given an initialization for the camera model $\boldsymbol{c}_{\text{init}} = [s_{\text{init}}, \boldsymbol{t}_{\text{init}}]$, LMR uses the concatenated vector $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_t, \boldsymbol{\beta}_t, \boldsymbol{c}_{\text{init}}]$ as part of its camera recurrent model to predict the camera model $\boldsymbol{c}_t, t = 1, \ldots, T$ for each frame. Note that while we have simplified the discussion and notation

here for clarity of exposition, LMR actually processes each batch of input in an iterative fashion, which we next describe in more mathematical detail.

Training an LMR model

As noted above and in Figure 6.3, the proposed LMR module takes as input the video feature set $\mathbf{\Phi}$ and the mean pose and shape parameters $\boldsymbol{\theta}_{mean}$ and $\boldsymbol{\beta}_{mean}$ and produces the set of parameter vectors $\mathbf{\Theta}_t = [\boldsymbol{\theta}_t, \boldsymbol{\beta}_t, \boldsymbol{c}_t]$ for each frame t. The LMR block processes each input set in an iterative fashion, with the output after each iteration being used as a new initialization point to further refine the result. The final output $\mathbf{\Theta}_t$ is then obtained at the end of L such iterations. Here, we provide further details of this training strategy.

Let each iteration step above be denoted by the letter v. At step $v = 0$, the initial pose and shape values for frame t will then be $\boldsymbol{\theta}_{t,v} = \boldsymbol{\theta}_{\text{mean}}$ and $\boldsymbol{\beta}_{t,v} = \boldsymbol{\beta}_{\text{mean}}$. The t, v notation refers to the v^{th} iterative step of LMR for frame number t. So, given $\mathbf{\Phi}$, $\boldsymbol{\beta}_{t,v}$, and the root pose $\boldsymbol{\theta}^1_{t,v}$ (recall root is part number 1 from above), the input to the root RNN will be the set of t vectors $[\mathbf{\Phi}_t, \boldsymbol{\theta}^1_{t,v}, \boldsymbol{\beta}_{t,v}]$ for each of the t frames. The root RNN then estimates an intermediate residual pose $\Delta\boldsymbol{\theta}^1_{t,v}$, which is added to the input $\boldsymbol{\theta}^1_{t,v}$ to give the root RNN output $\boldsymbol{\theta}^1_{t,v} = \boldsymbol{\theta}^1_{t,v} + \Delta\boldsymbol{\theta}^1_{t,v}$.

Given the root prediction $\boldsymbol{\theta}^1_{t,v}$ at iteration v, each of the other dependent part RNNs then use this information to produce their corresponding pose outputs. Specifically, for part RNN k, the input vector set (across the t frames) will be $[\mathbf{\Phi}_t, \boldsymbol{\theta}^k_{t,v}, \boldsymbol{\beta}_{t,v}, \boldsymbol{\theta}^1_{t,v}]$ for $k = 2, \ldots, 6$. Each part RNN first gives its corresponding intermediate residual pose $\Delta\boldsymbol{\theta}^k_{t,v}$. This is then added to its corresponding input part pose, giving the outputs $\boldsymbol{\theta}^k_{t,v} = \boldsymbol{\theta}^k_{t,v} + \Delta\boldsymbol{\theta}^k_{t,v}$ for $k = 2, \ldots, 6$.

After producing all the updated pose values at iteration $v = 0$, LMR then updates the shape values. Recall that the shape initialization used at $v = 0$ is $\boldsymbol{\beta}_{t,v} = \boldsymbol{\beta}_{\text{mean}}$. Given $\mathbf{\Phi}$, the

updated and aggregated pose vector set $\boldsymbol{\theta}_{t,v} = [\boldsymbol{\theta}^1_{t,v}, \ldots, \boldsymbol{\theta}^6_{t,v}]$, and the shape vector set $\boldsymbol{\beta}_{\text{mean}}$, LMR then uses the input vector set $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_{t,v}, \boldsymbol{\beta}_{\text{mean}}]$ as part of the shape update module to produce the new shape vector set $\boldsymbol{\beta}_{t,v}$ for each frame t during the iteration v.

Given these updated $\boldsymbol{\theta}_{t,v}$ and $\boldsymbol{\beta}_{t,v}$, LMR then updates the camera model parameters (used for image projection) with a camera model RNN. We use an RNN to model the camera dynamics to cover scenarios where the camera might be moving, although a non-dynamical fully-connected neural network can also be used in cases where the camera is known to be static. Given an initialization for the camera model $\boldsymbol{c}_{t,v} = \boldsymbol{c}_{\text{init}}$ at iteration $v = 0$, the camera RNN processes the input vector set $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}_{t,v}, \boldsymbol{\beta}_{t,v}, \boldsymbol{c}_{\text{init}}]$ to produce the new camera model set $\boldsymbol{c}_{t,v}$ for each frame t.

After going through one round of pose update, shape update, and camera update as noted above, LMR then re-initializes this prediction process with the updated pose and shape vectors from the previous iteration. Specifically, given the updated $\boldsymbol{\theta}_{t,v}$ and $\boldsymbol{\beta}_{t,v}$ at the end of iteration $v = 0$, the root RNN at iteration $v = 1$ then takes as input the set $[\boldsymbol{\Phi}_t, \boldsymbol{\theta}^1_{t,v}, \boldsymbol{\beta}_{t,v}]$, where the pose and shape values are not the mean vectors (as in iteration $v = 0$) but the updated vectors from iteration $v = 0$. LMR repeats this process for a total of V iterations, finally producing the parameter set $\boldsymbol{\Theta}_t = [\boldsymbol{\theta}_t, \boldsymbol{\beta}_t, \boldsymbol{c}_t]$ for each frame t. Note that this iterative strategy is similar in spirit to the iterative error feedback strategies commonly used in pose estimators [107, 305, 49, 193].

All the predictions above are supervised using several cost functions. First, if ground-truth SMPL model parameters $\boldsymbol{\Theta}^{gt}_t$ are available, we enforce a Euclidean loss between the predicted and the ground-truth set:

$$L_{\text{smpl}} = \frac{1}{T} \sum_{t=1}^{T} \|\boldsymbol{\Theta}^{gt}_t - \boldsymbol{\Theta}_t\|_2 \tag{6.2}$$

where the summation is over the $t = T$ input frames in the current batch of data.

Next, if ground-truth 3D joints $\boldsymbol{X}_t^{gt} \in \mathbb{R}^{3\times K}$ (recall K=24 from Section 6.2.4) are available, we enforce a mean per-joint L1 loss between the prediction 3D joints $\boldsymbol{X}_t \in \mathbb{R}^{3\times K}$ and $\boldsymbol{X}_t^{gt}$. To compute $\boldsymbol{X}_t$, we use the predicted parameter set $\boldsymbol{\Theta}_t$ and the SMPL vertex mapping function $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) : \mathbb{R}^{82} \rightarrow \mathbb{R}^{3\times N}$ and the joint regression matrix $\boldsymbol{W}$ (see Section 6.2.4). The loss then is:

$$L_{\text{3D}} = \frac{1}{T}\frac{1}{K}\sum_{t=1}^{T}\sum_{k=1}^{K} \|\boldsymbol{X}_{k,t}^{gt} - \boldsymbol{X}_{k,t}\|_1 \tag{6.3}$$

where each column of $\boldsymbol{X}_{k,t}^{gt} \in \mathbb{R}^3$ and $\boldsymbol{X}_{k,t} \in \mathbb{R}^3$ is one of K joints in three dimensions and the outer summation is over $t = T$ frames as above.

Finally, to provide supervision for camera prediction, we also enforce a mean per-joint L1 loss between the prediction 2D joints $\boldsymbol{x}_t \in \mathbb{R}^{2\times K}$ and the ground-truth 2D joints $\boldsymbol{x}_t^{gt}$. To compute $\boldsymbol{x}_t$, we use the 3D joints prediction $\boldsymbol{X}_t$ and the camera prediction $\boldsymbol{c}_t$ to perform an orthographic projection following Equation 6.1. The loss then is:

$$L_{\text{2D}} = \frac{1}{T}\frac{1}{K}\sum_{t=1}^{T}\sum_{k=1}^{K} \|\boldsymbol{x}_{k,t}^{gt} - \boldsymbol{x}_{k,t}\|_1 \tag{6.4}$$

where each column $\boldsymbol{x}_{k,t}^{gt} \in \mathbb{R}^2$ and $\boldsymbol{x}_{k,t} \in \mathbb{R}^2$ of $\boldsymbol{x}_t^{gt}$ and $\boldsymbol{x}_t$ respectively is one of K joints on the image and the outer summation is over $t = T$ frames as above.

The overall LMR training objective then is:

$$L_{\text{LMR}} = w_{\text{smpl}}L_{\text{smpl}} + w_{\text{3D}}L_{\text{3D}} + w_{\text{2D}}L_{\text{2D}} \tag{6.5}$$

where w_{smpl}, w_{3D}, and w_{2D} are the corresponding loss weights.

6.4 Experimental Results

6.4.1 Datasets and Evaluation

Following Kocabas *et al.* [215], we use a mixture of both datasets with both 2D (e.g., keypoints) as well as 3D (e.g., mesh parameters) annotations. For 2D datasets, we use PennAction [481], PoseTrack [9], and InstaVariety [194], whereas for 3D datasets, we use Human3.6M [182], MPI-INF-3DHP [295], and 3DPW [416]. In all our experiments, we use exactly the same settings as Kocabas *et al.* [215] for a fair benchmarking of the results. To report quantitative performance, we use evaluation metrics that are now standard in the human mesh research community. On all the test datasets, we report both mean-per-joint position error (MPJPE) as well as Procrustes-aligned mean-per-joint position error (PA-MPJPE). Additionally, following Kanazawa *et al.* [194] and Kocabas *et al.* [215], on the 3DPW test set, we also report the acceleration error ("Accel."), which is the average (across all keypoints) difference between the ground truth and predicted acceleration of keypoints, and the per-vertex error (PVE). All implementation details and hyperparameters are provided in the supplementary material.

6.4.2 Ablation Results

Table 6.1: Results of an ablation study comparing LMR with different number of frames for temporal modeling. No. Frames: sequence length along the temporal dimension.

Methods	No. Frames	Human3.6M		MPI-INF-3DHP	
		MPJPE↓	Rec. Error↓	MPJPE↓	Rec. Error↓
LMR	4	65.8	45.1	99.2	66
LMR	8	64.4	44.7	97.52	64.83
LMR	16	**61.9**	**42.5**	**94.6**	**62.4**

We first present results of an ablation experiment conducted to study how the length of input video sequence during temporal modeling will affect the performance of the **LMR**. We follow the same pipeline as Figure 3, with the only difference using different numbers of frames as the input. Specifically, we use 4, 8 and 16 as the length of video sequence. We show quantitative results on Human3.6M [182] and MPI-INF-3DHP [295] in Table 6.1, where gradual improvements are observed by increasing sequence length for temporal modeling and using 16 frames of input video sequence in LMR gives better performance. Next, we present both per-part and per-action evaluations of LMR on the Human3.6M [182] dataset in Table 6.2 and Table 6.3, where one can note improved performance of LMR on both per-body-part and per-action-label basis.

Table 6.2: Per-part evaluation on Human3.6M.

Body parts	HMR		LMR	
	MPJPE ↓	Rec. Error ↓	MPJPE ↓	Rec. Error ↓
Right Leg	73.3	38.7	**55.7**	**27.2**
Left Leg	72.8	38.9	**57.3**	**24.9**
Right Arm	116.4	54.9	**79.1**	**40.9**
Left Arm	111.2	56.1	**79.7**	**40.5**
Head	75.5	49.3	**64.1**	**40.7**

Table 6.3: Per-action reconstruction error (lower is better) on Human3.6M.

Actions	HMR	LMR	Actions	HMR	LMR
Directions	54.6	**40.5**	Discussion	58.5	**43.1**
Eating	60.3	**45.5**	Greeting	59.6	**44.5**
Phoning	66.2	**44.9**	Photo	76.9	**46**
Posing	59.2	**39.9**	Purchases	54.7	**38**
Sitting	73.4	**56.4**	SittingDown	80.7	**68.8**
Smoking	63.3	**47.9**	Waiting	58.5	**39.5**
WalkDog	64.9	**41.6**	Walking	50.7	**33.8**
WalkTogether	54.6	**37.8**			

Table 6.4: Results of an ablation study comparing LMR with a single RNN baseline.

Methods	Human3.6M		MPI-INF-3DHP		3DPW			
	MPJPE↓	Rec. Error↓	MPJPE↓	Rec. Error↓	MPJPE↓	Rec. Error↓	PVE↓	Accel↓
LMR no root	66.7	43.5	97.1	64	86.3	55.1	98.9	17.6
Full model	**61.9**	**42.5**	**94.6**	**62.4**	**81.7**	**51.2**	**93.6**	**15.6**

Finally, we study the impact of LMR's root dependency design choice. Here, we follow the same pipeline as Figure 3 but without the use of root dependencies in inferring the pose parameters of the non-root body parts. From Table 6.4, one can note the full proposed model, i.e., including root dependencies, gives better performance than the design without using root dependencies, validating our design choice.

6.4.3 Comparison with the state of the art

We compare the performance of LMR with a wide variety of state-of-the-art image-based and video-based methods. We first begin with a discussion on relative qualitative performance. In Figure 6.4, we show three frames from two different video sequences in (a) and (b) comparing the performance of the image-based HMR method [193] (first row) and our proposed LMR. Since LMR is a video-based method, one would expect substantially better performance, including in cases where there are self-occlusions. From Figure 6.4, one can note this is indeed the case. In the first column of Figure 6.4, HMR is unable to infer the correct head pose (it infers front facing when the person is actually back back facing), whereas LMR is able to use the video information from prior to this frame to infer the head pose correctly. Note also HMR's incorrect inference in other local regions, e.g., legs, in the subsequent frames in Figure 6.4(a). This aspect of self-occlusions (i.e., invisible face keypoints) is further demonstrated in Figure 6.4(b), where HMR is unstable (front

facing on a few and back facing on a few frames), whereas LMR consistently infers the correct pose.

Next, we compare the performance of LMR with the state-of-the-art video-based VIBE method [215]. In Figure 6.5, we show three frames from two different video sequences in (a) and (b). One can note substantial performance improvement in several local regions from these results. In particular, LMR infers more accurate hand pose and camera model parameters in Figure 6.5(a) when compared to VIBE. The results in Figure 6.5(b), a more challenging scenario, best illustrates the benefits offered by proposed local design of LMR. Given the variety of body movements in this set of frames, one can note the improved performance of LMR in several regions- hands and legs in the first column, head in the second column, and hands and legs again in the third column. These results are further substantiated in the quantitative comparison we discuss next.

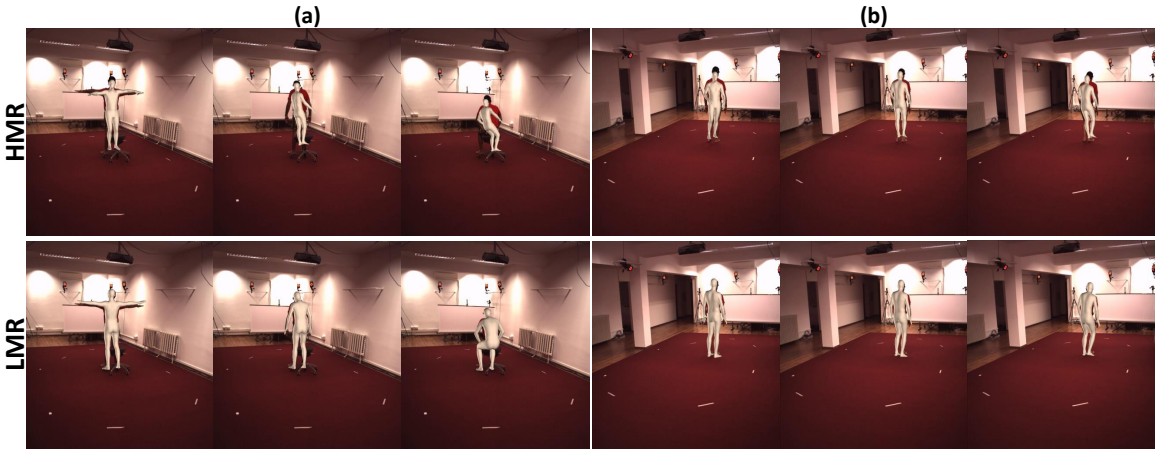

Figure 6.4: Two sets of qualitative results comparing the performance of LMR with the image-based HMR [193] method.

We provide a quantitative comparison of the performance of LMR to various state-of-the-art image- and video-based methods in Table 6.5. We make several observations. First, as

expected, LMR gives substantially better performance when compared to the image-based method of Kanazawa *et al.* [193] (MPJPE of 61.9 mm for LMR vs. 88.0 mm for HMR on Human3.6M, 94.6 mm for LMR vs. 124.2 mm for HMR on MPI-INF-3DHP, and 81.7 mm for LMR vs. 130.0 mm for HMR on 3DPW). This holds with other image-based methods as well (first half of Table 6.5). Next, LMR gives competitive performance when compared to state-of-the-art video-based methods as well. In particular, further substantiating the discussion above, LMR generally outperforms Kocabas *et al.* [215] with margins that are higher on the "in-the-wild" datasets (MPJPE of 94.6 mm for LMR vs. 96.6 mm for Kocabas *et al.* [215] on MPI-INF-3DHP, Accel. of 15.6 mm/s^2 for LMR vs. 23.4 mm/s^2 for Kocabas *et al.* [215] on 3DPW), further highlighting the efficacy of LMR's local dynamic modeling.

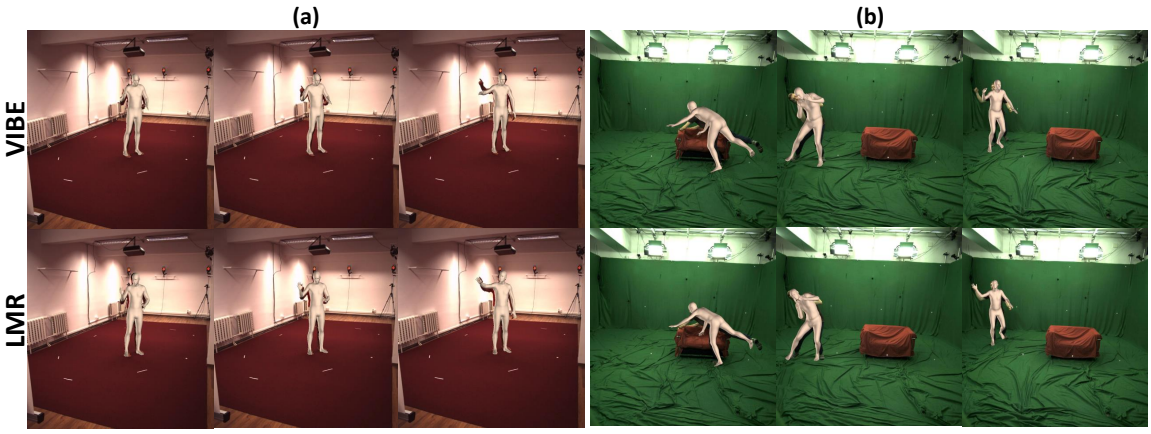

Figure 6.5: Two sets of qualitative results comparing the performance of LMR with the video-based VIBE [215] method.

Finally, in Table 6.5, we also compare our results with those of Kolotouros *et al.* [216] that uses an additional step of in-the-loop model fitting. Note that despite our proposed LMR **not** doing this extra model fitting, it outperforms Kolotouros *et al.* [216] in most cases, with particu-

larly substantial performance improvements on MPI-INF-3DHP (MPJPE of 94.6 mm for LMR vs. 105.2 mm for Kolotouros *et al.* [216]) and 3DPW (MPJPE of 81.7 mm for LMR vs. 96.9 mm for Kolotouros *et al.* [216]).

Table 6.5: Comparing LMR to the state of the art ("-": unavailable result in the corresponding paper).

Methods		Human3.6M		MPI-INF-3DHP		3DPW			
		MPJPE ↓	Rec. Error ↓	MPJPE ↓	Rec. Error ↓	MPJPE ↓	Rec. Error ↓	PVE ↓	Accel ↓
Image-based	Kanazawa *et al.* [193]	88.0	56.8	124.2	89.8	130	76.7	-	37.4
	Omran *et al.* [307]	-	59.9	-	-	-	-	-	-
	Pavlakos *et al.* [317]	-	75.9	-	-	-	-	-	-
	Kolotouros *et al.* [217]	-	50.1	-	-	-	70.2	-	-
	Georgakis *et al.* [133]	67.7	50.1	-	-	-	-	-	-
Extra-fitting	Kolotouros et al. [216]	62.2	**41.1**	105.2	67.5	96.9	59.2	116.4	29.8
Video-based	Kanazawa *et al.* [194]	-	56.9	-	-	116.5	72.6	139.3	**15.2**
	Arnab *et al.* [13]	77.8	54.3	-	-	-	72.2	-	-
	Doersch *et al.* [106]	-	-	-	-	-	74.7	-	-
	Kocabas *et al.* [215]	65.6	41.4	96.6	64.6	82.9	51.9	99.1	23.4
	LMR	**61.9**	42.5	**94.6**	**62.4**	**81.7**	**51.2**	**93.6**	15.6

Chapter 7

Towards Visually Interpreting Variational Autoencoders

7.1 Introduction

As computer vision models, driven by deep learning [222, 163], begin to be successfully transitioned to real-world tasks like healthcare and robotics [191, 233], applications in these areas demand a clear understanding of the algorithm's reasoning apart from performance robustness. This has led to much recent interest in methods that help understand and explain the underlying *what* is driving the output and *why*.

Following the visualization method of Zeiler and Fergus [465], there has been substantial recent progress in designing techniques to visualize CNN feature activations. One such line of work uses the concept of CNN network attention [490, 369], typically described by means of attention maps that show which features are considered by a trained model to be "important" for satisfying

the criteria used to train the model. With these methods, one can generate attention maps that visualize object regions, e.g., a *cat*, that help in interpreting why the model classified the image to the *cat* category. Other extensions [239, 422, 485] propose methods that use these attention maps as explicit trainable constraints and show improved downstream performance and visual interpretability. However, a key limitation of both class-activation-mapping (CAM)-type methods [490, 369] as well as recent extensions [239, 422, 485] is the need for an explicit classification module, typically achieved by means of cross-entropy-type losses, to generate these attention maps, limiting most of their use to problems involving classification/categorization.

Given the aforementioned progress in interpreting classification models, one would naturally like to explain a wider variety of computer vision models. For instance, while progress in algorithms for generative modeling has been rapid [212, 145, 294, 185, 430, 445, 277, 286], there is still much room for research in visually interpreting these models. While there is ongoing work in *using* the concept of visual attention in generative models [392, 6, 472], the focus here is on using network attention as an auxiliary information source (e.g., for training), and not to interpret/explain the underlying model.

In this chapter, we present methods that help take a step towards bridging the aforementioned gap. In particular, we consider VAEs [212] as an instantiation of generative models and propose techniques to generate attention maps, called *VAE attention*, that we seek to use to interpret the latent space learned by a VAE. It is important to note that our core ideas are not limited to VAEs and can be extended to other classes of generative models as well, e.g., GANs [145]. The key intuition underlying our VAE attention is that the latent space encapsulates all the properties of a

trained VAE and that computing attention maps conditioned on these latent variables will help interpret any downstream VAE predictions.

Given that a trained VAE that represents a distribution over all the latent variables, we seek to model VAE attention by considering not just a single sample but also the entire learned distribution. To this end, we propose two ways to calculate and generate VAE attention: *sample attention* and *distribution attention*. Given an input image, our methods first infer a latent space distribution from the encoder of the trained VAE. Given this distribution, *sample attention* samples a latent code and computes derivatives (with respect to this code) of the encoder's convolutional feature maps at the layer immediately preceding fully connected units, which are then aggregated into the visual attention map for this input image. While this method can help interpret the VAE from the perspective of a single sample of the latent space, there is typically much more information in the entire distribution that what a single sample can capture. To help take a step towards interpreting this distribution, our proposed *distribution attention* collects a number of latent codes (by repeated sampling) from the distribution, generates per-code sample attention maps, and fits a pixel-wise Gaussian distribution. The resulting "mean" and "standard deviation" attention maps help visually interpret not only the prediction confidence of the VAE but also the associated uncertainty.

While these visual attention maps help interpret the latent space (which we show with qualitative results), we show how they can be used in many more creative ways. For instance, a classical VAE application involves localizing anomalies, where the idea is, in the learned latent space, any input not from the Gaussian distribution used to train the VAE is anomalous. Our VAE attention maps can naturally be used to "interpret" this behavior by conditioning our attention computation process on the anomalous latent codes. These attention maps can also be used as cues to go

a step further and precisely localize the anomaly in the image. We conduct extensive experiments on multiple natural and industrial image datasets and present state-of-the-art anomaly localization results with standard VAE models (i.e., no bells and whistles) while also generating attention maps that help interpret predictions.

Another key research topic in the study of VAEs is in disentangling the learned latent space, and there has been much recent progress here as well [170, 208, 486]. Our VAE attention maps can naturally be used to improve the disentanglement performance. The key idea is once we compute VAE attention maps conditioned on the learned latent space, they can serve as explicit trainable constraints during the model learning process. To this end, we present a new learning objective called attention disentanglement loss and show how it can be used in conjunction with existing/standard VAE models to improved latent space disentanglement (as measured by standard evaluation metrics). We demonstrate this by means of extensive experiments on the Dsprites [293] and Fashion-MNIST [447] datasets.

7.2 Related Work and Our Contributions

7.2.1 CNN Visual Explanations.

The work of Zeiler and Fergus [465] and Mahendran and Vedaldi [288] have been widely adopted to visualize intermediate CNN feature layers for understanding the activity within the layers of convolutional nets. Some more recent visual-attention-based approaches along this line of work can be categorized into either gradient-based methods or response-based methods, including [490, 129, 369, 58, 98, 310]. GradCAM [369] is a widely used gradient-based method which computes and visualizes gradients backpropagated from the decision output to a convolutional feature

layer. On the other hand, response-based approaches and its applications [477, 490, 129] typically compute the attention maps by means of extending the original CNN architecture with additional trainable units. The goal of both lines of work is to localize attentive and informative image regions that contribute the most to the model prediction. However, these methods and their extensions [129, 239, 422], while being able to explain classification/categorization models well, cannot be trivially extended to explain deep generative models such as VAEs. In this work, we propose techniques to compute and visualize attention maps directly from the learned latent embedding of the VAE. Furthermore, we make the visual attention maps end-to-end trainable and show how such a change can result in improved latent space disentanglement.

7.2.2 Anomaly Detection.

Unsupervised anomaly detection [4] is still a challenging computer vision task. Classification-based [352, 55] or reconstruction-based approaches are two main streams used for anomaly detection in much recent work. Classification-based approaches aim to progressively learn representative one-class decision boundaries like hyperplanes [55] or hyperspheres [352] around the normal-class input distribution to tell outliers/anomalies apart. However, these methods have difficulty dealing with high-dimensional data [54]. Reconstruction-based models, on the other hand, are proposed based on the hypothesis that input data that are anomalous cannot be reconstructed well by a model that is trained only with normal input data. This principle has been used by several methods based on the traditional PCA [209], sparse representation [484], and more recently deep autoencoders [501, 491] and generative models [494, 480]. In this work, we take a different approach to tackle this problem. We use the attention maps generated by our proposed VAE attention generation method as cues to localize anomalies. Our intuition is that representations of anomalous data should be

reflected in latent embedding as being anomalous, and that generating input visual interpretations from such an embedding will give us the information we need to localize the particular anomaly.

7.2.3 VAE Disentanglement Learning.

Kingma *et al.* [212] proposed a VAE with a stochastic variational inference and learning algorithm in generative modelings for large dataset. Recently, a number of following work has been expended in understanding latent space disentanglement for generative models. Schmidhuber *et al.* [362] proposed a principal algorithm to disentangle latent representations by minimizing the predictability of one latent dimension given other dimensions. Desjardins *et al.* [99] generalized an approach to factor the latent variables. Chen *et al.* [67] designed the InfoGAN to maximise the mutual information between a subset of latent variables and the observation. Esmaeili *et al.* [116] introduced hierarchically factorized VAEs. β-VAE [170] attempted to explore independent latent factors of variation in observed data. While still a popular unsupervised framework, β-VAE sacrificed reconstruction quality for obtaining better disentanglement. Chen *et al.* [64] extended β-VAE to β-TCVAE by introducing a total correlation-based objective, whereas Mathieu *et al.* [292] explored decomposition of the latent representation into two factors for disentanglement, and Kim *et al.* [208] proposed FactorVAE that encouraged the distribution of representations to be factorial and independent across the dimensions. Xiang *et al.* [446] proposed a multi-factor disentanglement learning framework for multi-conditional generation in a weakly-supervised manner. Yang *et al.* [453] proposed CausalVAE to transform independent exogenous factors for disentanglement learning. While these methods concentrate on factorizing the latent representations provided by each individual latent neuron, we take a different approach. We enforce learning a disentangled space by formulating disentanglement constraints based on our proposed visual interpretations, *i.e.*,

visual attention maps. To this end, we propose a new attention disentanglement learning objective that we quantitatively show, provides superior performance when compared to existing work.

7.2.4 Contributions of this Chapter

- We take a step towards addressing the relatively unexplored problem of visually interpreting variational autoencoders, presenting techniques to generate visual attention maps, called VAE attention, conditioned on latent codes sampled from the latent space of a trained VAE.
- We demonstrate two different ways to generate VAE attention: sample attention which are generated simply from samples of latent vectors and distribution attention which are computed from distributions of sampled latent vectors in latent space in a statistical manner.
- We show how VAE attention can be put to multipurpose use apart from interpreting the latent space.
  - First, we demonstrate the use of VAE attention in localizing anomalies in images. We show both qualitative results with VAE attention maps and state-of-the-art quantitative performance on multiple natural, industrial, and medical image datasets.
  - Next, we show how the proposed VAE attention maps lead to a new learning objective, called the attention disentanglement loss, that can be used in conjunction with standard VAE models. We demonstrate it leads to improved disentanglement performance by means of extensive experiments on standard benchmark datasets.

7.3 Technical Approach

In this section, we present gradient-based attention map generating approaches towards visually interpreting a variational autoencoder (VAE). In Section 7.3.1, we first give a brief review of the VAE architecture. In Section 7.3.2, we introduce our technique to generate *sample attention* maps from samples of latent vectors, to visually interpret learned VAE latent space given in-/out-domain data points. In Section 7.3.3, we take a step forward and generate *distribution attention* maps given distributions of sampled latent vectors inferred from input data, for more generic and comprehensive interpretations of a learned VAE statistically. Finally in Section 7.3.4, we discuss about how to apply our generated VAE attention maps to different applications.

7.3.1 Variational Autoencoder

A vanilla variational autoencoder (VAE) is typically trained with the standard reconstruction loss between the input and the decoded/reconstructed data, along with a variational objective term which attempts to learn a standard normal latent space distribution. The Kullback-Leibler distribution metric, which is typically computed between the latent space distribution and the standard Gaussian distribution, is chosen as the variational objective in VAEs. To be more specific, given the input data $\mathbf{x}$, the conditional distribution $q(\mathbf{z}|\mathbf{x})$ of the encoder, the standard Gaussian distribution $p(\mathbf{z})$, and the reconstructed data $\hat{\mathbf{x}}$, a vanilla VAE aims to optimize:

$$L = L_r(\mathbf{x}, \hat{\mathbf{x}}) + L_{\text{KL}}(q(\mathbf{z}|\mathbf{x}), p(\mathbf{z})) \tag{7.1}$$

where L_{KL} is the Kullback-Leibler divergence term and L_r is the reconstruction term, which is generally a mean-squared error between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

7.3.2 Interpreting VAEs with Sample Attention

Our first method, called *sample attention* (SA), for interpreting a VAE is based on calculating gradients of prediction vectors with respect to convolutional feature maps. Our ideas are different from the existing work [369, 490, 485] that compute attention maps by backpropagating the score from a classification model. Instead, in our work, the attention maps generated with SA are computed directly from the learned latent space. As illustrated in Fig. 7.1 and discussed below, we compute a score from the latent space, which is then used to calculate gradients and obtain the attention map.

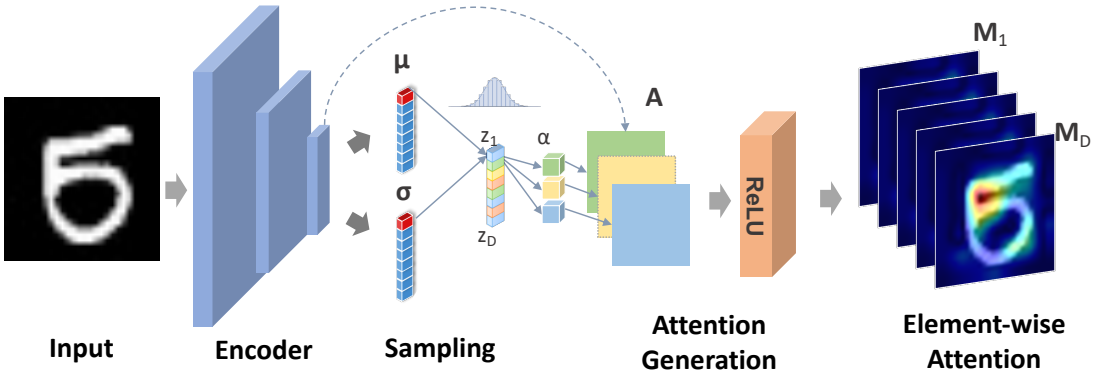

Figure 7.1: Element-wise attention generation and aggregation with a VAE.

Given the posterior distribution $q(\mathbf{z}|\mathbf{x})$ inferred by the trained VAE for a data sample $\mathbf{x}$, we sample a latent sample $\mathbf{z}$ from the posterior distribution during the testing time. For each element z_i in $\mathbf{z}$, we backpropagate gradients to the last chosen convolutional feature maps $\mathbf{A} \in \mathbb{R}^{n \times h \times w}$ to generate the attention map $\mathbf{M}_i$ corresponding to z_i. Specifically, $\mathbf{M}_i$ is computed as the linear combination:

$$\mathbf{M}_i = \text{ReLU}(\sum_{k=1}^{n} \alpha_k \mathbf{A}_k) \tag{7.2}$$

where the scalar $\alpha_k = \text{GAP}(\frac{\partial z_i}{\partial \mathbf{A}_k})$ and $\mathbf{A}_k$ is the k^{th} feature channel ($k = 1, \ldots, n$) of the feature maps $\mathbf{A}$. Note $\frac{\partial z_i}{\partial \mathbf{A}_k}$ is a matrix and so we use the global average pooling (GAP) operation to get the scalar α_k:

$$\alpha_k = \frac{1}{T} \sum_{p=1}^{h} \sum_{q=1}^{w} (\frac{\partial z_i}{\partial A_k^{pq}}) \tag{7.3}$$

where $T = h \times w$, and A_k^{pq} is the pixel value at location (p, q) of the $h \times w$ matrix $\mathbf{A}_k$. We now repeat this procedure for all the elements $z_1, z_2, \ldots, z_D$ in the $D-$dimensional latent vector $\mathbf{z}$ to yield $\mathbf{M}_1, \ldots, \mathbf{M}_D$ (see Fig. 7.1). Such a procedure gives one attention map $\mathbf{M}_i$ per latent element. Yet, one can obtain a single overall attention map by adopting any matrix aggregation scheme. For instance, we take the mean of all the element-wise attention maps so the overall attention map is $\mathbf{M} = \frac{1}{D} \sum_i^D \mathbf{M}_i$ which is referred as the VAE attention. We denote such aggregated VAE attention map $\mathbf{M}$ as a *sample attention* (SA) map.

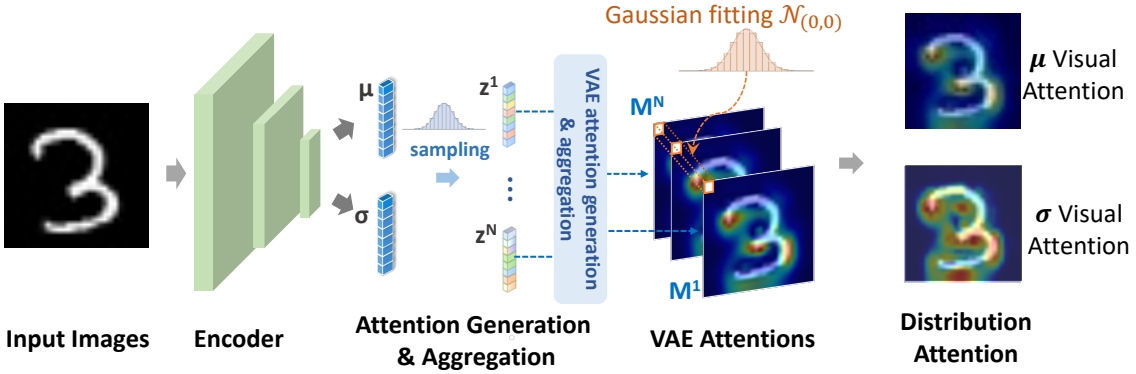

Figure 7.2: Generating the distribution of visual attentions with a VAE.

7.3.3 Interpreting VAEs with Distribution Attention

In the previous section, we presented a technique to generate VAE *sample attention* from *one sample* latent code of the learned latent space distribution.

However, since SAs are computed from individual latent codes, they do not help to accurately interpret the entire distribution. To address this issue, we present our second method which is to generate attention maps directly from the distribution, called *distribution attention* (DA).

Specifically, from the mean and standard deviation vectors $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ learned by the VAE , we sample N latent codes $\mathbf{z}^j, j = 1, \ldots, N$ and generate the corresponding SAs $\mathbf{M}^j, j = 1, \ldots, N$. Next, for each pixel location (u, v) in the SA maps, we fit and estimate a Gaussian distribution with these N SA maps values $\mathbf{M}^j_{(u,v)}, j = 1, \ldots, N$:

$$\mathcal{N}^{\mathbf{M}}_{(u,v)} \sim (\mu^{\mathbf{M}}_{(u,v)}, \sigma^{\mathbf{M}}_{(u,v)}) \tag{7.4}$$

where $0 \leq u < H$ and $0 \leq v < W$ and $\mathcal{N}^{\mathbf{M}}_{(u,v)}$ is the normal distribution for the pixel location (u, v). By estimating $\mathcal{N}^{\mathbf{M}}_{(u,v)}$ for all the pixel locations, we can obtain a $\mu^{\mathbf{M}}$ map and a $\sigma^{\mathbf{M}}$ map, as *distribution attention* maps. This process is visually summarized in Fig. 7.2.

Our proposed DA maps help provide both a statistical visualization of a VAE's learned latent space while also capturing a more comprehensive and diverse interpretation of the latent distribution when compared to SA maps.

7.3.4 Applications of VAE Attention

While both SA and DA attention maps help to interpret a VAE by means of the learned latent space (which we discuss both qualitatively and quantitatively in the following sections), they

can be put to multipurpose use. First, in Section 7.4, we show how one can use our methods to localize anomaly regions given a trained one-class VAE. Our hypothesis is a one-class VAE trained with "normal" data will infer a distribution that "deviates" from the "normal" distribution when presented with "abnormal" data at test time. We validate this hypothesis by first presenting a new anomaly localization framework that solely relies on VAE attention maps and then extensively evaluate framework on a variety of image datasets from different domains - natural images and industrial images.

Furthermore, in section 7.5 we introduce a new method that uses and applies VAE attentions in learning and improving the disentanglement of the latent feature space. To this end, we propose a new learning objective using VAE attention that facilitates feature space disentanglement and then show how it can be integrated with a standard VAE architecture. We demonstrate efficacy by means of an extensive set of experiments on various standard disentanglement learning datasets.

7.4 Unsupervised Anomaly Localization

Given a one-class VAE trained as described in Section 7.3.1, our key ideas are as follows. If we input a test sample from the "normal" class (e.g., digit "1" if the VAE is trained with samples from this class), the inferred latent space will ideally conform to the standard normal distribution. Consequently, given a testing sample from a different class (digit "7" as an example), the latent space inferred by the VAE should deviate considerably from the distribution learned with data from the normal class. This intuition can be formalized mathematically with Gaussian functions by means of both our proposed methods: *sample attention* and *distribution attention*. In the following sections,

we will elaborate on our approaches, as well as the quantitative and qualitative results on a variety of image datasets across multiple domains.

7.4.1 Anomaly Localization with Sample Attention

Under the aforementioned train/test setting for the anomaly localization task using an one-class VAE, a VAE is trained to approximately model the probability density of the distribution of the training set. During inference time, a latent code that deviates far away from the learned normal data distribution is hence seen as an anomaly. To better represent and study such deviation, we define a normal difference distribution (NDD) function described in the following.

Given the normal data $\mathbf{x} \in \mathbf{X}$ used to train the VAE, one can infer the overall $\mu^{\mathbf{x}}$ and $\sigma^{\mathbf{x}}$ of the normal data distribution. When an abnormal sample $\mathbf{y} \in \mathbf{Y}$ is fed to the trained VAE for testing, a latent code $\mathbf{z_i}$ is inferred from the $\mu_{\mathbf{i}}^{\mathbf{y}}$ and $\sigma_{\mathbf{i}}^{\mathbf{y}}$, where i is the number of samples. Thus, we define the NDD of each latent code $\mathbf{z_i}$ in the latent space as:

$$P_{q(\mathbf{z_i}|\mathbf{x})-q(\mathbf{z_i}|\mathbf{y})}(u) = \frac{e^{-[u-(\mu^{\mathbf{x_i}}-\mu^{\mathbf{y_i}})]^2/[2((\sigma^{\mathbf{x_i}})^2+(\sigma^{\mathbf{y_i}})^2)]}}{\sqrt{2\pi((\sigma^{\mathbf{x_i}})^2+(\sigma^{\mathbf{y_i}})^2)}} \tag{7.5}$$

For clarity, in the following we will denote the normal difference distribution as $\mathcal{N}^{ndd} \sim (\boldsymbol{\mu}^{ndd}, \boldsymbol{\sigma}^{ndd})$. Following our SA generation procedure introduced in Section 7.3.2, we can now compute an anomaly attention map $\mathbf{M}$ corresponding to a random sampled latent vector $\mathbf{z}$ from the normal difference distribution $P_{q(\mathbf{z}|X)-q(\mathbf{z}|Y)}$. This process is visually summarized in the top branch of Fig. 7.3.

In Fig. 7.4, we show two sets of example results for anomaly SA maps. The first set includes results from the object class *hazelnut* where we train a VAE using all the "normal" (non-defect) *hazelnut* images. An example "normal" image is shown on the top row of the figure for

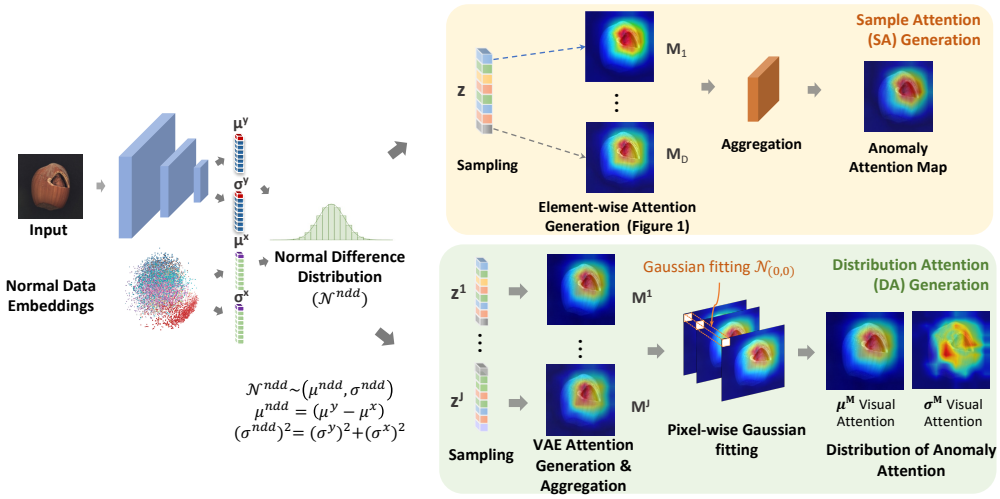

Figure 7.3: Attention generation with a one-class VAE. Top branch: generating anomaly attention as the Sample Attention (see Section 7.4.1). Bottom branch: visualizing the distribution of anomaly attention as the Distribution Attention (see Section 7.4.2).

clarity. During testing, inputs include "abnormal" *hazelnut* images with different types of defects, such as holes, cracks, or prints (sample images of different defects are shown from left to right in the middle row of Fig. 7.4). We present our anomaly SA maps on the bottom row, where high-response regions are in red and correspond to anomalies in the given images. One can note these attention maps can localize defects accurately. Similar observations can be made in the second set of results for the *leather* object class as well.

To further illustrate how the sampled latent code corresponds to and characterizes the computed SA maps, we generate multiple samples of latent vectors from $\mathcal{N}^{ndd}$ by *traversing* the distribution from from very close to the mean ($\boldsymbol{\mu}^{ndd}$) to very far from it. We then compute the SA map corresponding to each sampled latent code and visualize it. Concretely, we expect that a latent vector $\mathbf{z}$ sampled close to $\boldsymbol{\mu}^{ndd}$ will yield an SA map with higher density of high-response regions (darker red) over anomalous pixels, as opposed to cases when we sample a $\mathbf{z}$ far away from the

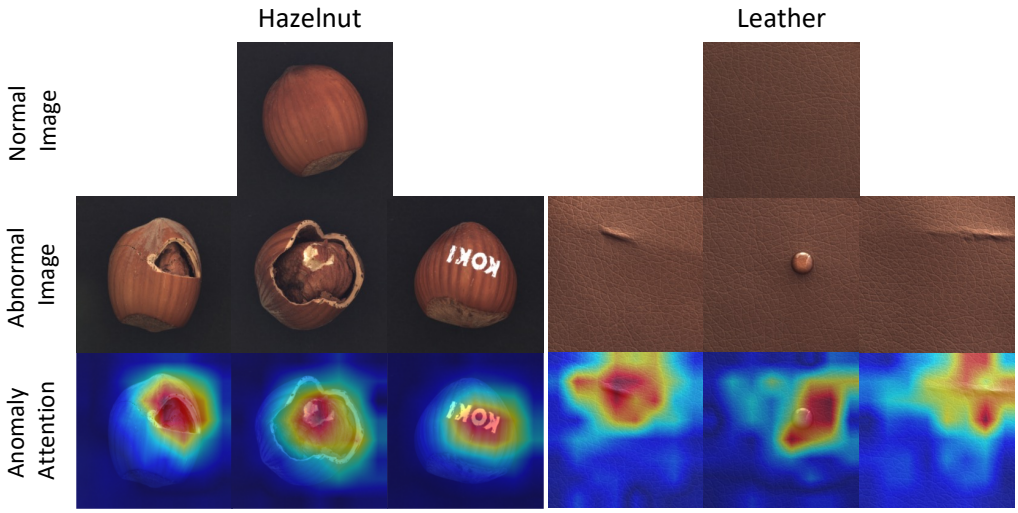

Figure 7.4: Anomaly sample attention maps generated for two different classes of objects: Hazelnut and Leather. On the top row, we show one sample image of the "normal" data each per class. In the middle row, for each class we show three different types of "abnormal" sample images (images with different types of detects on the objects). While on the bottom row, the corresponding generated SA maps are shown. In these examples, our attention maps correctly localize the anomalous regions (defects) within each testing images.

$\boldsymbol{\mu}^{ndd}$. We show results in Fig. 7.5, where one can note that we traverse the distribution by sampling latent vectors from far to close to the mean (from left to right as shown in the figure), the highlighted areas in the SA maps also gradually focus on the appropriate anomalous pixel regions in the image.

7.4.2 Anomaly Localization with Distribution Attention

We next discuss how one can use our proposed DA maps for localizing anomalies. As discussed in Section 7.3.3, we start by computing a large number of J anomaly SA maps $\mathbf{M}^j, j \in J$ from latent vectors $\mathbf{z}^j, j \in J$ randomly sampled from $\mathcal{N}^{ndd}$. Then at each pixel location (u, v) of these J SAs, we fit and estimate a pixel-wise Gaussian distribution with a mean of $\mu^{\mathbf{M}}_{(u,v)}$ and a standard deviation of $\sigma^{\mathbf{M}}_{(u,v)}$. Finally, we represent these pixel-wise Gaussian representations as two separate attention maps for visualization: one contains only values of $\mu^{\mathbf{M}}_{(u,v)}$ and the other those

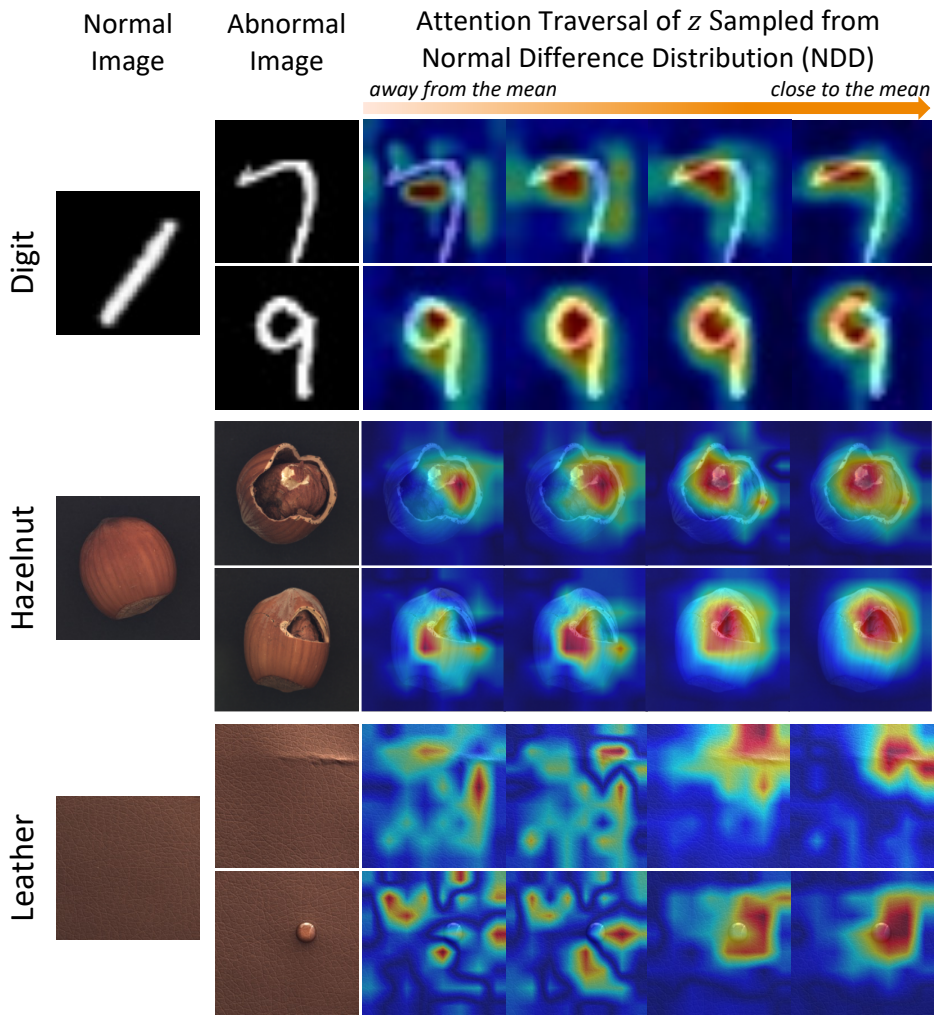

Figure 7.5: Attention traversal by using different $\mathbf{z}$ sampled from $\boldsymbol{\mu}^{ndd}$ given the abnormal data. From the left to the right we show attention maps of different latent codes $\mathbf{z}$ sampled from far to close to the $\boldsymbol{\mu}^{ndd}$. We can see that the abnormal regions are gradually highlighted precisely.

of $\sigma^{\mathbf{M}}_{(u,v)}$. We refer to them as mean anomaly DA map $\mu^{\mathbf{M}}$ and deviation anomaly DA map $\sigma^{\mathbf{M}}$ respectively. This process is visually summarized in the bottom branch of Fig. 7.3.

As a result, each pixel in $\mu^{\mathbf{M}}$ is a statistical summary of all the anomaly SA map responses at the same pixel location. Thus, if high responses at (u, v) are present across SA maps, then $\mu^{\mathbf{M}}_{(u,v)}$ will also have a high response, suggesting the pixel (u, v) is anomalous. On the other

hand, each pixel in $\sigma^{\mathbf{M}}$ represents the statistical uncertainty of the resultant SA maps' responses at this location. In other words, if uncertainty levels are high at (u, v) across SA maps, then $\sigma^{\mathbf{M}}_{(u,v)}$ will have a high response as well. This means the uncertainty of the accuracy of the $\mu^{\mathbf{M}}_{(u,v)}$ value is high. Consequently, the DA maps $\mu^{\mathbf{M}}$ and $\sigma^{\mathbf{M}}$ taken together provide a more comprehensive interpretation of the anomaly localization predictions of our framework.

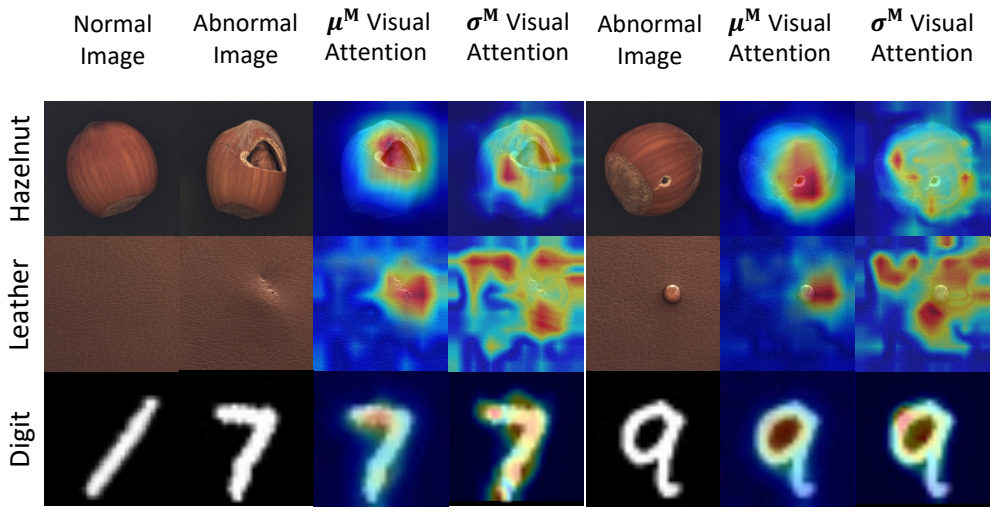

Figure 7.6: $\mu^{\mathbf{M}}$ and $\sigma^{\mathbf{M}}$ visual attention maps of the distribution of anomaly attention over the abnormal images. The $\mu^{\mathbf{M}}$ visual attention maps present the average likelihood of each pixel being detected as anomaly and $\sigma^{\mathbf{M}}$ visual attention maps present the uncertainty of each pixel being detected as anomaly.

In Fig. 7.6, we show some examples of DA maps. For instance, in the first row, we present two examples from *hazlenut* object class. The second through fourth columns show the input (anomalous) *hazelnut* image, the $\mu^{\mathbf{M}}$ map, and the $\sigma^{\mathbf{M}}$ map, respectively. Fifth through seventh columns show another example and its corresponding DA maps. In the first example on the first row, one can note high responses in $\mu^{\mathbf{M}}$ are at the `crack` pixel regions in the image, whereas high responses in $\sigma^{\mathbf{M}}$ are present at the edges of the `crack` pixels. The interpretation from these

results is that while the model confidently predicts pixels inside the `crack` region to be anomalous, it is relatively less confident about pixels at the edges. This result is aligned with our intuition that it is generally challenging to tell if these boundary pixels are normal or abnormal, hence leading to higher uncertainty. Similar observations can also be made from other examples as well (including from the digits figures). This discussion shows how our proposed μ^{M} and σ^{M} pair of DA maps together help both visually localize anomalies as well as interpret the results.

7.4.3 Experiments for Anomaly Localization

In this section, we exploit SA and DA generation techniques over multiple datasets for the anomaly localization problem. We will show extensive experimental results both qualitatively and quantitatively.

Metrics

AUROC: We adopt the commonly used ,area under the receiver operating characteristic curve (ROC AUC), for evaluation of all quantitative anomaly detection performance evaluations. We define the true positive rate (TPR) as the percentage of samples that are correctly classified as anomalous across the whole testing class, and the false positive rate (FPR) as the percentage of samples that are wrongly classified as anomalous. For image-level performance, the AUROC value is calculated based on the anomaly score generated for each testing sample. For pixel-level performance, the AUROC value is generated based on attention maps and ground truth masks. For each given method, higher AUROC values indicate better performance at detecting anomalies.

Best IOU: In addition, in order to compare fairly against the other existing methods on the MVTec-AD dataset, we also compute the best intersection-over-union (IOU) score by searching for the best threshold based on the ROC curve.

Datasets

MNIST: The MNIST database[96] of handwritten digits has a training set of 60,000, and a test set of 10,000 examples partitioned into 10 digit classes. The images have been size-normalized and centered into 28×28 size.

UCSD Ped1 Dataset: UCSD Ped 1[245] pedestrian video dataset contains videos captured with a stationary camera to monitor a pedestrian walkway. This dataset includes 34 training sequences and 36 testing sequences, with about 5500 "normal" frames and 3400 "abnormal" frames. We resize the data to 100×100 pixels for training and testing.

MVTec-AD Dataset: MVTec-AD[23] is a real-world industrial image anomaly detection dataset with 5354 high-resolution images in 15 categories. The training set has 3629 normal images, and the test set contains 1725 normal or abnormal images. The ground-truth in the test set includes both labels and anomaly masks. We follow the original dataset split of MVTec-AD, i.e., use only anomaly-free images in training, and test on both normal and abnormal images.

Evaluation of Anomaly Localization with Sample Attention

In this sub-section, we present experiments for anomaly localization using VAE attention maps computed with the *sample attention* technique proposed in Section 7.4.1. Please see additional

qualitative results in the supplementary materials. We also conduct image-level anomaly classification experiments, results for which are shown in supplementary material.

MNIST

To prove the validity of our attention map generation methods, we first conduct a qualitative sanity test on the MNIST dataset [96]. All training and testing images are at resolution of 28×28 pixels. We train a one-class VAE model using the training images of one digit only and test on images of other digits from the testing set. For instance, we pick digit "1" as our training set to train a VAE. Then we could feed a digit "7" during testing. We expect that our approach will generate attention maps showing the anomalous regions of a "7" different from a "1".

More qualitative results are shown in Fig. 7.7. For instance, with the top-left example, we trained a VAE model with images of digit "1" as the normal class and test on images of all other digits as abnormal classes. For each test image, we infer the latent vector from the encoder of the trained VAE model and generate the attention map by sampling latent codes from the normal difference distribution. As can be observed from the results, the attention maps computed with the proposed method is intuitively satisfying. For example, with test data from class "7" as abnormal samples, our intuition is that a key difference between the "1" and the "7" is the top-horizontal bar in "7", and our generated attention map indeed highlights this region. One can make similar observations from the results of other digits as well (e.g.,"3").

UCSD Ped 1

We next evaluate our proposed anomaly localization technique on the UCSD Ped 1[245] pedestrian video dataset. We first show qualitative results in Fig. 7.8. In each row (left to right), the first two images are the input and corresponding ground-truth anomaly masks (e.g., *bicycle*, *Car* etc.),

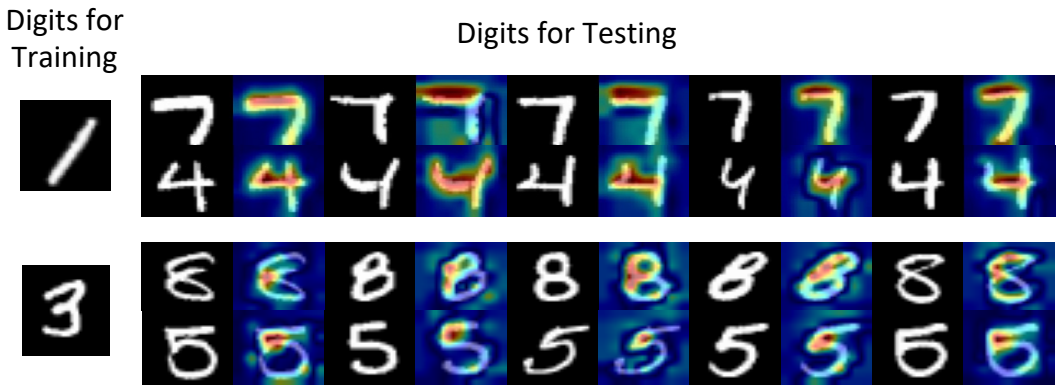

Figure 7.7: Anomaly localization results from the MNIST dataset.

the next two images are our results (attention map and mask), and the last two images show the difference maps *diffmap* between the input and its VAE reconstruction output. We note more precise localization of the high-response regions in our generated attention maps, and these high-response regions indeed correspond to anomalies in these images.

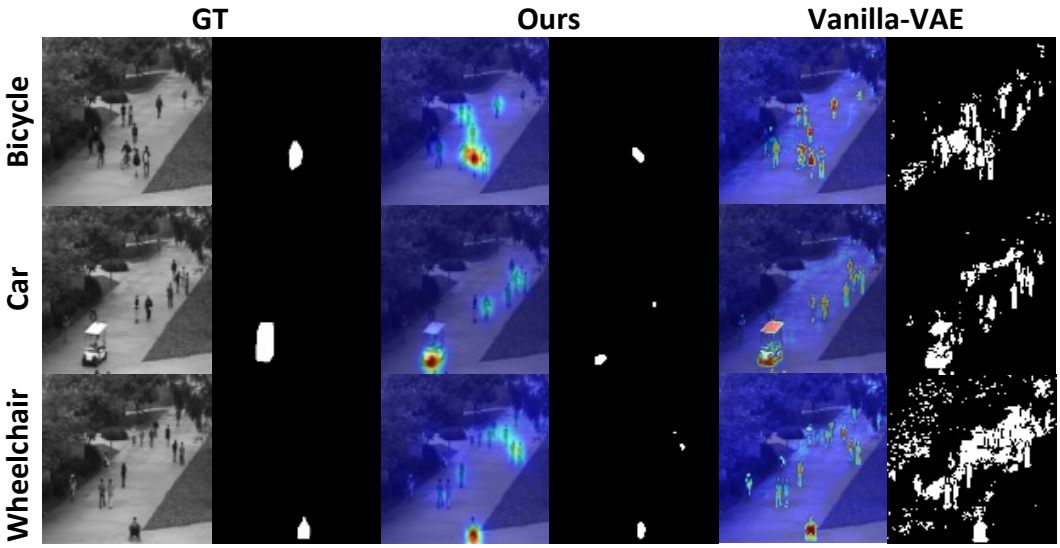

Figure 7.8: Qualitative results from UCSD Ped1 dataset. L-R: Original test image, ground-truth masks, our anomaly attention localization maps, and difference between input and the VAE's reconstruction. The anomalies in these samples are moving cars, bicycle, and wheelchair.

We also conduct experiments and ablation studies for pixel-level segmentation AUROC score against the baseline method of difference between input data and the reconstruction output. We test our proposed attention generation mechanism with varying levels of spatial resolution by backpropagating from the latent space to each of the encoder's convolutional layers: 50×50, 25×25, and 12×12. The quantitative results in Table 7.1 show our method gives better performance when compared to the baseline.

Table 7.1: Results on UCSD Ped1 using pixel-level segmentation AUROC score. We compare results obtained using our anomaly attention generated with different target network layers to reconstruction-based anomaly localization using Vanilla-VAE. SA: Sample Attention (see Section 7.4.1).

	diffmap	Ours(SA, Conv1)	Ours(SA, Conv2)	Ours(SA, Conv3)
AUROC	0.86	0.89	**0.92**	0.91

MVTec-AD

In this section, we conduct extensive qualitative and quantitative experiments on MVTec-AD and summarize the results below.

We use the ResNet18 [163] architecture as the backbone for the feature encoder of the one-class VAE (with a 32-dimensional latent space). As in the original work [23], we also perform random mirroring and random rotation for data augmentation on the training set. Given a test image as the abnormal input, the *sample attention* map is generated as discussed in Section 7.4.1. After obtaining the attention maps, binary localization maps are computed by using a variety of thresholds on the pixel response values, which is encapsulated in the ROC curve. We then calculate the area under the ROC curve and the best IOU number is computed based on the FPR and TPR values from the ROC curve.

Table 7.2: Quantitative results for pixel level segmentation on 15 categories from MVTec-AD dataset. We adopt comparison scores from [428]. We compare to two sets of baselines. The first set includes **domain-general** methods: SSIM-AE[24], l_2-AE[155], AnoGAN[361], and CNN-FD[301]. The second set includes **domain-specific** methods: SMAI[250], GDR[92], P-Net[497], and PatchNet[428]. SA: Sample Attention (see Section 7.4.1).

	Category	SSIM-AE[24]	l_2-AE[155]	AnoGAN[361]	CNN-FD[301]	SMAI[250]	GDR[92]	P-Net[497]	PatchNet[428]	**Ours (SA)**
Texture	Carpet	0.87	0.59	0.54	0.72	0.88	0.74	0.57	**0.96**	0.78
	Grid	0.94	0.90	0.58	0.59	0.97	0.96	**0.98**	0.78	0.73
	Leather	0.78	0.75	0.64	0.87	0.86	0.93	0.89	0.90	**0.95**
	Tile	0.59	0.51	0.50	0.93	0.62	0.65	**0.97**	0.80	0.80
	Wood	0.73	0.73	0.62	0.91	0.80	0.84	**0.98**	0.81	0.77
Object	Bottle	0.93	0.86	0.86	0.78	0.86	0.92	**0.99**	0.93	0.87
	Cable	0.82	0.86	0.78	0.79	0.92	0.91	0.70	**0.94**	0.90
	Capsule	**0.94**	0.88	0.84	0.84	0.93	0.92	0.84	0.90	0.74
	Hazelnut	0.97	0.95	0.87	0.72	0.97	**0.98**	0.97	0.84	**0.98**
	Metal Nut	0.89	0.86	0.76	0.82	0.92	0.91	0.79	0.84	**0.94**
	Pill	0.91	0.85	0.87	0.68	0.92	0.93	0.91	**0.93**	0.83
	Screw	0.96	0.96	0.80	0.87	0.96	0.95	**1.00**	0.96	0.97
	Toothbrush	0.92	0.93	0.90	0.77	0.96	**0.99**	**0.99**	0.96	0.94
	Transistor	0.90	0.86	0.80	0.66	0.85	0.92	0.82	**1.00**	0.93
	Zipper	0.88	0.77	0.78	0.76	0.90	0.87	0.90	**0.99**	0.78

The quantitative results are shown in Tables 7.2 and 7.3, where the performance of our proposed method is evaluated using the same evaluation techniques as in the benchmark paper of Wang *et al.* [428] (note that the baselines here are the same methods as in [23, 428]). We compare to two sets of baselines. The first set includes domain-general methods: SSIM-AE[24], l_2-AE[155], AnoGAN[361], and CNN-FD[301]. The second set includes domain-specific methods: SMAI[250], GDR[92], P-Net[497], and PatchNet[428]. From Table 7.2, one can note that the results of our method are competitive with respect to these methods, including better performance on some categories. **It is worth noting here that most of these methods are specifically designed for the anomaly localization task with task-specific network designs, whereas we train a standard VAE and generate our VAE attention maps for anomaly localization. Despite this simplicity, our method achieves competitive performance, demonstrating the potential of such an attention generation technique to be useful for tasks other than just model interpretation**.

Table 7.3: Quantitative results of best IOU for pixel level segmentation on 15 categories from MVTec-AD dataset. SA: Sample Attention (see Section 7.4.1).

	Category	SSIM-AE	l_2-AE	AnoGAN	CNN-FD	**Ours (SA)**
Texture	Carpet	0.69	0.38	0.34	0.2	0.1
	Grid	0.88	0.83	0.04	0.02	0.02
	Leather	0.71	0.67	0.34	0.74	0.24
	Tile	0.04	0.23	0.08	0.14	**0.23**
	Wood	0.36	0.29	0.14	0.47	0.14
Object	Bottle	0.15	0.22	0.05	0.07	**0.27**
	Cable	0.01	0.05	0.01	0.13	**0.18**
	Capsule	0.09	**0.11**	0.04	0.00	**0.11**
	Hazelnut	0.00	0.41	0.02	0.00	**0.44**
	Metal Nut	0.01	0.26	0.00	0.13	**0.49**
	Pill	0.07	0.25	0.17	0.00	0.18
	Screw	0.03	0.34	0.01	0.00	0.17
	Toothbrush	0.08	0.51	0.07	0.00	0.17
	Transistor	0.01	0.22	0.08	0.03	**0.30**
	Zipper	0.10	0.13	0.01	0.00	0.06

We also show some qualitative results in Fig. 7.9 (top to bottom: original image, ground truth segmentation mask, our method's anomaly attention map) on a wide range of categories, including both textures and objects. For each category, we show four types of defects provided by the dataset. One can note that our attention maps are able to accurately localize anomalous regions across these various defect categories.

Evaluation of Anomaly Localization with Distribution Attention

In this section, we present anomaly localization experimental results using the proposed *distribution attention* (DA) on MVTec-AD, UCSD Ped 1 and ISIC 2019 datasets. For each dataset, we follow the same training settings introduced in Section 7.4.3. During the testing, instead of using attention maps from SA (Section 7.4.1), we utilize the DA maps proposed in the Section 7.4.2 (see Fig. 7.3 bottom branch) for more accurate anomaly detection. To elaborate, after obtaining the $\mu^{\mathbf{M}}$ and $\sigma^{\mathbf{M}}$ given the input abnormal images, we use the $\sigma^{\mathbf{M}}$ map to weight the $\mu^{\mathbf{M}}$ map and generate the final anomaly detection result, *i.e.*, $\widetilde{\mu}^{\mathbf{M}} = \mu^{\mathbf{M}} * \frac{1}{1+\exp(-\alpha*(\sigma^{\mathbf{M}}-\beta))}$, α, β are

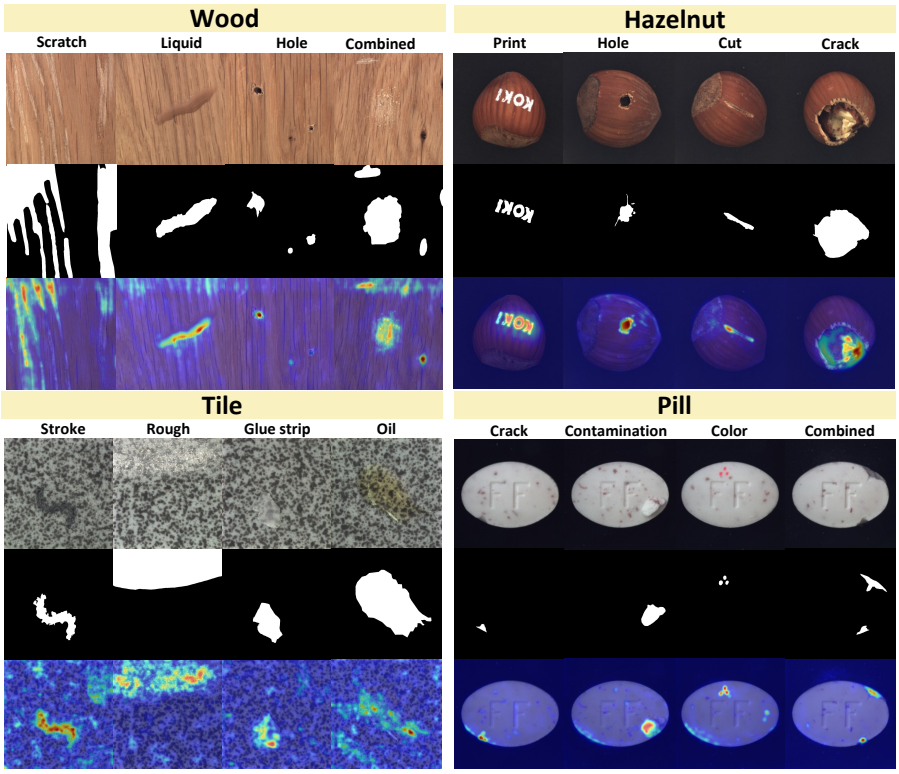

Figure 7.9: Qualitative results from MVTec-AD for Wood, Tile, Hazelnut and Pill. For each category there are four different type of defects. Our anomaly attention maps are able to accurately localize anomalies.

predefined hyperparameters set by cross-validation in all our experiments. Our expectation is that the high response regions (which also indicates high anomaly uncertainties) in σ^{M} visual attention can help remove uncertainties in the μ^{M} visual attention so that refined $\widetilde{\mu}^{\mathrm{M}}$ can detect anomalies with better accuracy and less uncertainty as well.

MVTec-AD

We firstly present quantitative results in Table 7.4 and where the performance of our proposed method is evaluated using the same evaluation techniques in [428]. We compare the results of the proposed method using the SAs and DAs. From Table 7.4, it is observed that our anomaly localization approach using the proposed *distribution attention* can consistently improve anomaly

localization performance across most of categories, which proves that rather than using only *one sample* from the latent space (NDD), considering the collections of attention maps generated with both mean and standard deviation from the learned distribution (NDD) can further help interpret the VAE model and in turn improve anomaly detection results.

Table 7.4: Quantitative results and comparisons for pixel level segmentation on 15 categories from MVTec-AD dataset. SA: Sample Attention (see Section 7.4.1). DA: Distribution Attention (see Section 7.4.2).

	Carpet	Grid	Leather	Tile	Wood	Bottle	Cable	Capsule	Hazelnut	Metal Nut	Pill	Screw	Toothbrush	Transistor	Zipper
Ours (*SA*)	0.78	0.73	0.95	0.8	0.77	0.87	0.9	0.74	0.98	0.94	0.83	0.97	0.94	0.93	0.78
Ours (*DA*)	**0.8**	**0.75**	**0.96**	0.8	**0.81**	**0.89**	**0.93**	0.74	0.98	0.94	0.83	**0.98**	**0.95**	0.93	**0.82**

UCSD Ped 1

We next conduct experiments on the UCSD Ped 1 dataset. We follow the same training settings introduced in Section 7.4.3 and use the *distribution attention* during the testing. We use the pixel-level segmentation AUROC score against the baseline method of difference between input data and the reconstruction output. Table 7.5 presents the quantatitive results between using our SA maps and DA maps, where we see using the proposed DA maps can consistently improve anomaly localization performance on the UCSD Ped 1 dataset.

Table 7.5: Results on UCSD Ped1 using pixel-level segmentation AUROC score.

	AUROC Score
Ours (Sample Attention)	0.92
Ours (Distribution Attention)	**0.95**

7.5 Disentangled Representation Learning

Learning disentangled representations with VAEs has been another intriguing area that has received much progress, as they can increase interpretability and generalization of the models and

learn faster on downstream tasks[279, 144, 406, 62, 208, 487]. A disentangled representation vector encodes isolated factors of variation in the input data into only a few independent dimensions[411]. We intuitively believe our proposed gradient-based VAE attention generated directly from the latent representation, can help visually explain latent space disentanglement. Furthermore, by leveraging these attention maps as part of trainable constraints, we are able to boost disentangled representation learning.

To this end, we will discuss our methodologies of applying our sample attention and distribution attention on disentanglement learning in the following sections. We will first present a novel learning objective called attention disentanglement loss (L_{AD}) and show how one can integrate it with existing frameworks, e.g., FactorVAE[208]. Next we will demonstrate the resulting impact in learning a disentangled embedding by means of qualitative attention maps and quantitatively performance characterization with standard disentanglement metrics on multiple datasets.

7.5.1 Attention-guided Representation Disentanglement

As discussed in Section 7.3, we proposed to generate VAE attention maps directly from each dimensions of a given latent vector. We have also shown that the resultant attention maps heat up certain regions (pixels) in the input image corresponding to the latent elements. Hence, we exploited these attention maps to provide visual clues to explain the VAE's latent representations. But here, we want to consider an alternative application to utilize our attention maps in the opposite direction. We seek to design a trainable constraints that attempts to separate the high-response regions in the attention maps generated from different latent vector's dimensions to be as much as possible. And by backpropagating from this attention-separation-driven constraints, the network explicitly forces the latent representation to be disentangled.

To follow this goal, we propose a new training loss called the *attention disentanglement* loss (L_{AD}). As mentioned, L_{AD} is designed to boost disentanglement learning by forcing attention separation. In other words, this loss penalizes on large number of overlapping high-response pixels within different attention maps. Given a pair of attention maps, $\mathbf{M}^p$ and $\mathbf{M}^q$, generated from different latent elements, we mathematically formulate L_{AD} as :

$$L_{AD} = 2 \cdot \sum_{p}^{D} \sum_{q,p \neq q}^{D} \frac{\sum_{ij} min(M_{ij}^p, M_{ij}^q)}{\sum_{ij} M_{ij}^p + M_{ij}^q} \tag{7.6}$$

D is the dimension of the latent vector, M_{ij}^p and M_{ij}^q are the $(i,j)^{th}$ pixel in the attention maps $\mathbf{M}^p$ and $\mathbf{M}^q$ respectively. One empirical implementation is taken as an example by considering two dimensions from the D where $p = 1$ and $q = 2$. Then two attention maps $\mathbf{M}^1$ and $\mathbf{M}^2$ can be computed from these two dimensions of latent vector, giving *attention disentanglement* loss (L_{AD}) of Equation 7.6 as:

$$L_{AD} = 2 \cdot \frac{\sum_{ij} min(M_{ij}^1, M_{ij}^2)}{\sum_{ij} M_{ij}^1 + M_{ij}^2} \tag{7.7}$$

Further, when considering three dimensions from the D of the latent vector, the *attention disentanglement* loss (L_{AD}) is calculated by iterating $p = 1, q = 2$, $p = 2, q = 3$ and $p = 1, q = 3$ so the Equation 7.6 is implemented as:

$$\begin{aligned} L_{AD} = 2 \cdot (&\frac{\sum_{ij} min(M_{ij}^1, M_{ij}^2)}{\sum_{ij} M_{ij}^1 + M_{ij}^2} + \frac{\sum_{ij} min(M_{ij}^2, M_{ij}^3)}{\sum_{ij} M_{ij}^2 + M_{ij}^3} + \\ &\frac{\sum_{ij} min(M_{ij}^1, M_{ij}^3)}{\sum_{ij} M_{ij}^1 + M_{ij}^3}) \end{aligned} \tag{7.8}$$

and L_{AD} achieves minimum value when not a single pair of attention maps have overlapping high-response pixels. Our proposed L_{AD} can be easily integrated with other learning objectives on existing VAE-type models. A better illustration of the training framework with our L_{AD} can be viewed in Fig. 7.10.

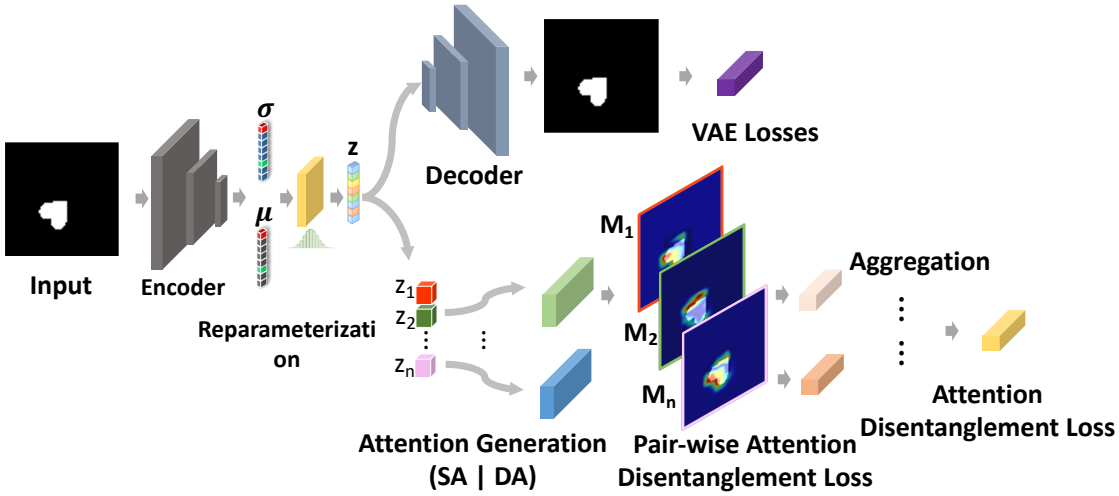

Figure 7.10: Training a variational autoencoder with the proposed attention disentanglement loss.

We then generate *sample attention* (SA) or *distribution attention* (DA) as the input attention maps for L_{AD}. In the case of SA, the procedure is straightforward: we choose a pair of attention maps from $\mathbf{M}_1, \ldots, \mathbf{M}_D$ (see Fig. 7.1) to compute the loss. While in the case of DA, we first sample N latent codes $\mathbf{z}^{\mathbf{k}}, k = 1, \ldots, N$, each with D dimension $\mathbf{z}^{\mathbf{k}} \in \mathbf{R}^D$. Then, for a chosen pair of dimensions *i.e.*, (p,q), we compute two sets of N element-wise attention maps $\mathbf{M_k}^p, k = 1, \ldots, N$ and $\mathbf{M_k}^q, k = 1, \ldots, N$ corresponding to p and q, respectively. Next, we fit two Gaussian distributions for these two sets and obtain two pairs of element-wise DA maps ($\boldsymbol{\mu}^{\mathbf{M}^p}$, $\boldsymbol{\sigma}^{\mathbf{M}^p}$) and ($\boldsymbol{\mu}^{\mathbf{M}^q}$, $\boldsymbol{\sigma}^{\mathbf{M}^q}$). When computing L_{AD}, we use $\boldsymbol{\sigma}^{\mathbf{M}^p}$ and $\boldsymbol{\sigma}^{\mathbf{M}^q}$ to weight the overlapped high-responsive regions following the same weighting function introduced in Section 7.4.3. Our expectations is that by using the *distribution attention*, attention maps generated from different dimensions should be separated as well as with low uncertainty.

Our proposed L_{AD} can be directly integrated with the existing network FactorVAE, and its standard training objective function L_{FV}. Our final overall learning objective is presented as

follows:

$$L = L_{\text{FV}} + \lambda L_{\text{AD}} \tag{7.9}$$

where λ is the hyperparameter to weight standard FactorVAE training loss and our attention disentanglement loss. Then we train FactorVAE with the overall learning objective Equation (7.9). With such explicit attention-separation constraints, we expect better disentanglement learning performance than the original FactorVAE pipeline. Note that while we use the FactorVAE [208] for demonstration in this work, the proposed attention disentanglement loss is in no way limited to this model and can be used in conjunction with other models as well (e.g., β-VAE [170]).

7.5.2 Evaluation of Disentangled Representation Learning

Disentanglement Metric

We adopt the disentanglement metric proposed in [208] to evaluate the disentanglement performance of learned VAE latent space.

Datasets

Fashion-MNIST dataset: The Fashion-MNIST dataset [447] comprises of a training set of 60,000 images and a test set of 10,000 images with the size of 28×28. Each image is in the grayscale format and associated with a label from 10 classes including T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot.

Dsprites dataset: The dSprites dataset [293] provides images of 2D shapes generated from 6 ground truth independent latent factors. These factors are color, shape, scale, rotation, x and y positions of a sprite. In total, the dataset contains 737,280 binary images with shape of 64×64.

Evaluation Results

Dsprites

We perform attention disentanglement experiment on Dsprites dataset [293] and present both qualitative and quantitative results. We start to implement our proposed method-attention disentanglement by considering two dimensions in the Equation 7.6 to calculate the *attention disentanglement* loss train the model using the same experimental settings described in the paper [208].

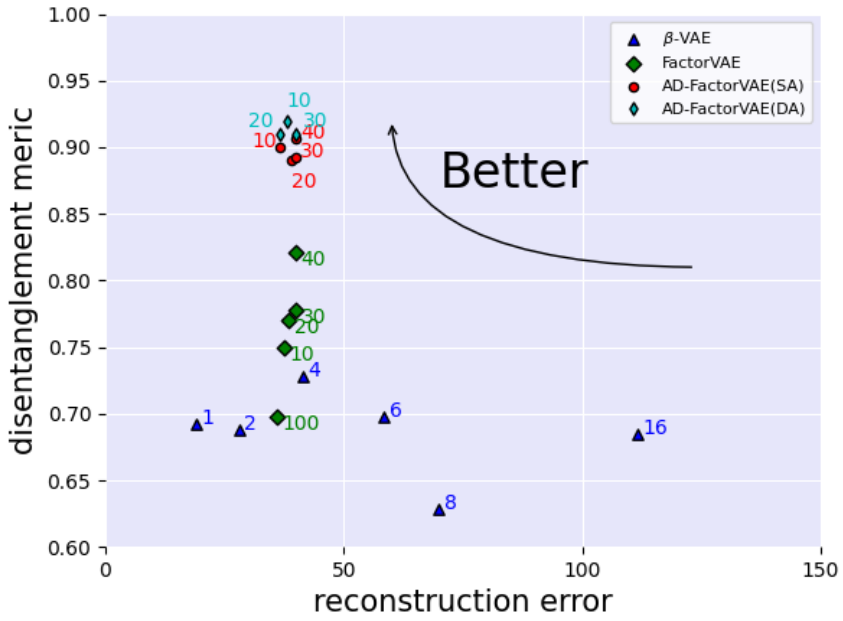

Figure 7.11: Reconstruction error plotted against disentanglement metric [208]. The numbers at each point show β and γ values. We want a low reconstruction error and a high disentanglement metric. AD-FactorVAE (SA): our proposed method of disentanglement learning with *sample attention* (see Section 7.3.2). AD-FactorVAE (DA): our proposed method of disentanglement learning with the *distribution attention* (see Section 7.3.3).

Fig. 7.11 presents the quantitative comparisons of the best disentanglement performance which is plotted against the reconstruction error between our proposed method (called AD-FactorVAE (SA)) with other competing methods: baseline FactorVAE [208] which is trained without attention

disentanglement loss and β-VAE[170]. From Fig. 7.11, it is observed that training the model by using overall learning objectives with our proposed disentanglement loss L_{AD} results in higher disentanglement scores under the same experimental settings. In more details, our method can give a best disentanglement score of around 0.90, whereas baseline FactorVAE ($\gamma = 40$) gives around 0.82, both with a reconstruction error around 40. Our proposed method also gives a higher disentanglement score compared to the best score of β-VAE which is 0.73 with $\beta = 4$. It is demonstrated that the potential of both our proposed VAE attention and L_{AD} in improving the performance of existing methods in the disentanglement literature.

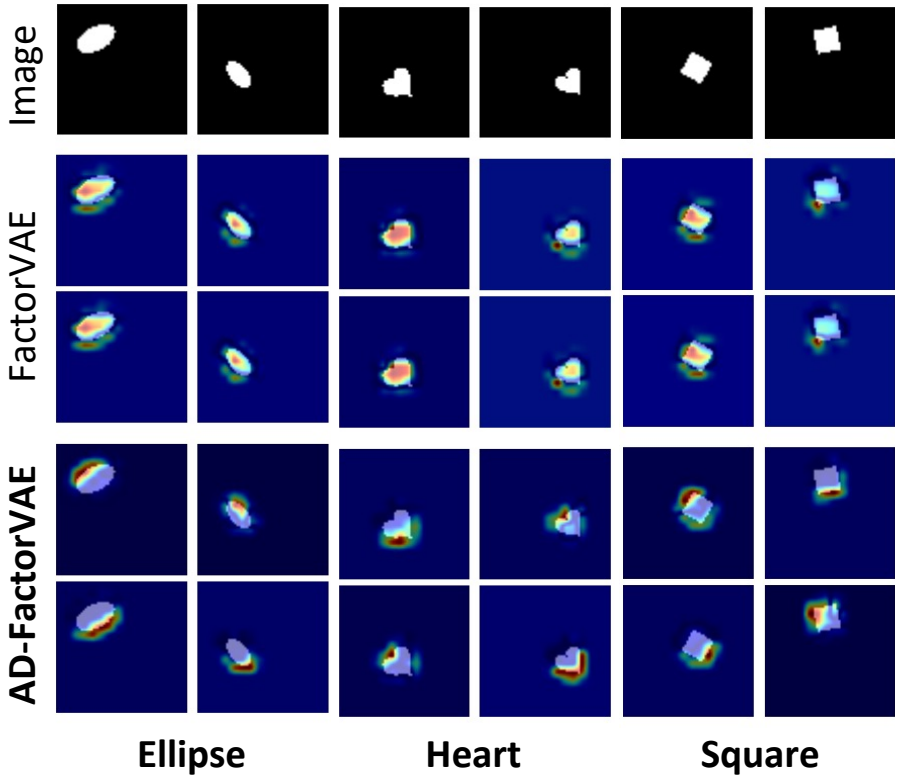

Figure 7.12: Attention separation on the Dsprites dataset. Top row: the original shape images. Middle two rows: attention maps from FactorVAE. Bottom two rows: attention maps from AD-FactorVAE.

Fig. 7.11 also presents experiments results for disentanglement learning with *distribution attention* (called AD-FactorVAE (DA)) on Dsprites dataset. During the training VAE, different from using *one sample* from the learned latent space to generate VAE attention, we generate the *distribution attention* maps $\mu^{\mathbf{M}}$ and $\sigma^{\mathbf{M}}$ after sampling a collection of latent codes and use these two attention maps to calculate the *attention disentanglement* loss (see Section 7.5.1). From the Fig. 7.11, it is observed that training the model with our proposed disentanglement loss L_{AD} using the DA maps can consistently improve results in higher disentanglement scores under the same experimental settings.

Fig. 7.12 shows some attention maps generated using the baseline FactorVAE and our proposed AD-FactorVAE. The first row shows 5 input images in different shapes, and the next 4 rows show attention results with the baseline FactorVAE and our proposed method. Row 2 shows attention maps generated with FactorVAE by backpropagating from the latent dimension with the highest response, whereas row 3 shows attention maps generated by backpropagating from the latent dimension with the next highest response. Rows 4 and 5 show the corresponding attention maps generated from the model trained with the proposed AD-FactorVAE. Our intuition and expectation with AD-FactorVAE is that each dimension's attention map will have high responses in different spatial regions of the input. From Fig. 7.12, we see that this is indeed the case where we observe that, high-response regions in different areas (by backpropogating different dimensions) of latent vectors in attention maps in row 4 and 5 from the proposed method AD-FactorVAE, whereas attentions are overlapped in row 2 and 3 where attention maps are generated from the baseline FactorVAE.

We further perform extensive attention disentanglement experiment on Dsprites dataset by considering more dimensions in implementing learning objectives with L_{AD}. We train the net-

work using our proposed method-attention disentanglement, rather than using a pair of attentions corresponding to two dimensions for training loss, we take additional 2 pairs of attentions corresponding 3 dimension in total for computing the learning objectives during the training. We use the same experimental settings described above, with different γ values as 10, 40. Both qualitative and quantitative results are presented and discussed below.

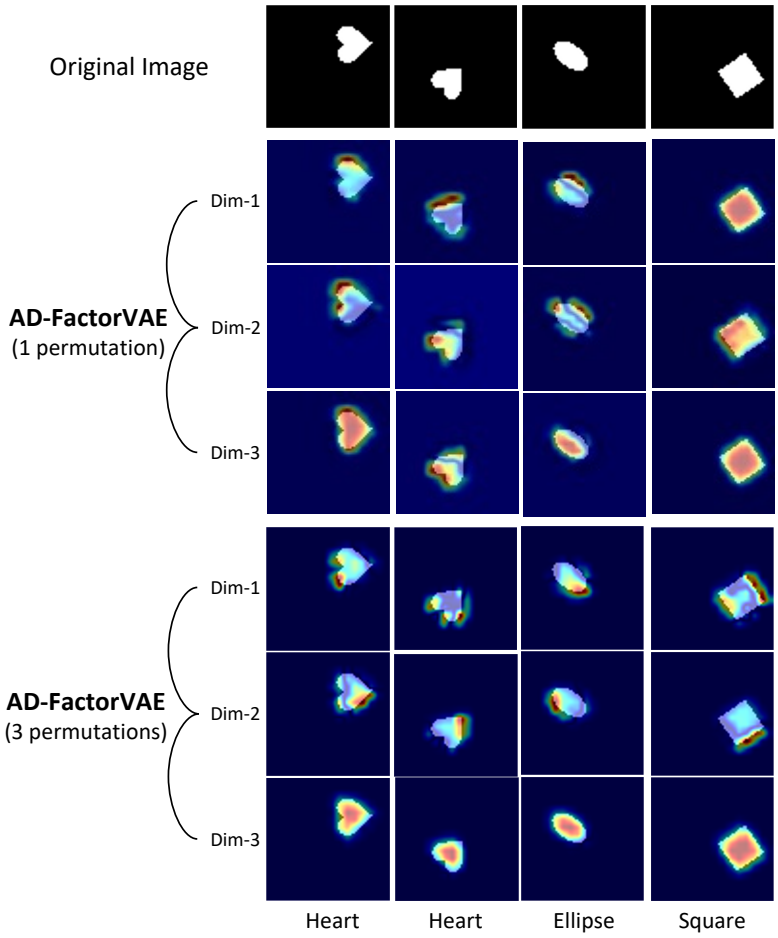

Figure 7.13: Attention separation on the Dsprites dataset. First row: the original shape images. Middle three rows: attention maps of three dimensions from AD-FactorVAE trained with 1 pair of attention map for attention disentanglement loss. Last three rows: attention maps generated from three different dimensions of latent vector from AD-FactorVAE trained with 3 pair of attention maps for attention disentanglement loss.

Fig. 7.13 shows some attention maps generated using the model trained by our proposed AD-FactorVAE. The first row shows input images in different shapes, and the next 3 rows show attention maps generated with our proposed method by considering 1 pair of attention maps in computing attention disentanglement loss. The last 3 rows show attention maps generated by back-propogating 3 dimensions of latent vector from the model trained with our proposed method by considering 3 pairs of attention maps in calculating attention disentanglement loss. From Fig. 7.13, better separated attention maps are observed in the last three rows, which meet our expectation that different spatial regions would be highlighted in attention maps generated from different dimensions of latent vector.

In Table 7.6, we present the best disentanglement performance of our proposed AD-FactorVAE by considering additional pairs of attention maps generated from different dimensions of latent vector when computing our attention disentanglement loss in training objectives. From Table 7.6, it is observed that training our AD-FactorVAE with additional dimensions in objectives can consistently improve disentanglement scores.

Table 7.6: Average disentanglement score on Dsprites Dataset using Kim *et al.* [208] metrics. The higher disentanglement score means the better disentanglement performance.

	γ=10	γ=40
AD-FactorVAE (1 pair of attentions)	0.90	0.91
AD-FactorVAE (3 pairs of attentions)	**0.93**	**0.93**

Chapter 8

Discussions and Future Work

This thesis proposed several novel applications for sports analytics, image and language pre-training, face synthesis, scene perceptions, human body mesh recovery and visual intepretations of variational autoencoder. In the second chapter, we proposed a novel framework for analyzing the fine-grained actions of soccer players in highlight videos of soccer games. We collected a dataset from various sources with detailed annotations that includes (a) two coarse-grained classes, dribbling and shooting, and (b) six fine-grained styles, three in dribbling – Stepover, Elastico and Chop and three in shooting – Penalty-Kick, Goal and Free-Kick. We performed extensive evaluation and comparisons of algorithms on the dataset and our proposed framework achieved the best performance in classifying soccer players' fine-grained actions. The proposed self-attention motion modelling module encoded fine-grained motion dynamics of a soccer player as auxiliary signals. The energy features and motion features are aggregated with the proposed energy-motion features aggregation module for fine-grained classification of a player's actions. In addition, we performed ablation experiments to evaluate the contributions of each component in the proposed framework,

and to classify fine-grained shooting actions of players and in return the defense actions; the defensive actions of goalkeepers have not been analyzed in existing work on analyzing the performance of soccer players in video. Future work will include extending the dataset to involve additional fine-grained actions of soccer games, and analyzing extensive fine-grained actions in sports videos, *e.g.*, players shooting directions, team offending/defending tactical styles and strategies.

In the third chapter, we presented the RECLIP, a method for resource-efficient language image pretraining. We proposed to leverage small images with paired texts for the main constrastive training phase and fine-tuned the model with high-resolution images for a short cycle at the end. The proposed training method has been validated on zero-shot image and text retrieval benchmarks and image classification datasets. In comparisons to the baseline method, RECLIP training recipe saves the computations by $6 \sim 8\times$ with improved zero-shot retrieval performance and competitive classification accuracy. Compared to the state-of-the art methods, RECLIP significantly saves $\mathbf{79}\% \sim \mathbf{98}\%$ resource in cores$\times$hours with highly competitive zero-shot classification and image-text retrieval performance. We hope RECLIP paves the path to make contrastive language image pretraining more resource-friendly and accessible to the broad research community. Future work will include integrating masking techniques in generating image tokens to reduce token length for the model training and evaluating a wider range of zero-shot down-streaming tasks.

In the fourth chapter, we streamlined the generation of visual interpretations by means of gradient-based attention mechanisms for facial attribute translation on the conditional generative adversarial networks and showed how they can be used during training to improve the performance of generative models. We developed a novel system called **AKD-GAN** where a teacher network distills its knowledge via visual attentions to a student network by proposing an attention knowledge

distillation loss in order to train the student network to generate better face images. We experimented with the proposed novel method with various system design implementations, including the teacher-student paradigms with different model complexities in student networks and attention knowledge distillation using full self-supervision and weak supervision for model training. We showed the effectiveness of the system in removing biases in the attribute generation. Finally, we evaluated the proposed framework on the widely used public face image datasets CelebA and human expression dataset RaFD, demonstrating with both quantitative and qualitative results the effectiveness of the approach for diverse applications. The proposed face synthesis with facial attributes translation can be applied in a wide range of computer vision and biometrics tasks, such as attribute-based recognition and identification, etc. Also, it can be applied as a powerful data augmentation tool by using synthesized face images with various facial attributes for multi-purposes usage, including large-scale training for deep models, adversarial attacks. etc. Besides, face images synthesized with various facial attributes can be potentially applied for disguised and concealed identity recognition. In addition, generating better face images with target attributes will have the potential impact for the entertainment industry and also for digital art. Finally, synthesizing facial attributes in videos for video generation in real time will be an interesting future work. Future work will include extending the proposed method on a wider range of GAN architecture and exploring advanced techniques to remove bias in generation faical images.

In the fifth chapter, a novel monocular self-supervised depth estimation framework, called the **MonoIndoor++**, has been proposed to predict depth map of a single image in indoor environments. The proposed model consists of three modules: (a) a novel *depth factorization module* with a transformer-based scale regression network which is designed to jointly learn a global depth

scale factor and a relative depth map from an input image, (b) a novel *residual pose estimation module* which is proposed to estimate accurate relative camera poses for novel view synthesis of self-supervised training that decomposes a global pose into an initial pose and one or a few residual poses, which in turn improves the performance of the depth model, (c) a *coordinates convolutional encoding module* which is utilized to encode coordinates information explicitly to provide additional cues for the residual pose estimation module. Comprehensive evaluation results and ablation studies have been conducted on a wide-variety of indoor datasets, establishing the state-of-the-art performance and demonstrating the effectiveness and universality of our proposed methods. Future work will include evaluating the proposed method in a supervised setup and training the proposed model on multiple datasets with various depth settings and then evaluating it for zero-shot cross-dataset transfer capabilities without fine-tuning.

In the sixth chapter, we considered the problem of video human mesh recovery and noted that the currently dominant design paradigm of using a single dynamical system to model all motion dynamics, in conjunction with a "flat" parameter regressor is insufficient to tackle challenging in-the-wild scenarios. We presented an alternative design based on local recurrent modeling, resulting in a structure-informed learning architecture where the output of each local recurrent model (representing the corresponding body part) is appropriately conditioned based on the known human kinematic structure. We presented results of an extensive set of experiments on various challenging benchmark datasets to demonstrate the efficacy of the proposed local recurrent modeling approach to video human mesh recovery.

In the seventh chapter, we proposed new techniques to visually interpret deep generative models, in particular, variational autoencoders, by means of gradient-based network attention. We

presented two ways to calculate visual attention maps. Firstly, unlike most the existing work which perform gradients backpropagation from the classification output, the proposed method computed gradients directly with a latent vector sampled from the latent space on VAEs and generated *sample attention* maps, which bridged the gap between generative models and gradient-base attention visualization mechanism. Secondly, further steps are taken to visually interpret the learned entire distribution, this work proposes to fit and estimate a pixel-wise Gaussian distribution of a collection of *sample attention* maps generated by sampling a group of latent codes and the resulting "mean" and "standard deviation" attention maps are presented as the *distribution attention* maps. Both two types of visual attention maps provided a explainable way to interpret generative models, they can also be used for down-stream computer vision tasks. In this work, two classical VAE applications are taken for experiments to demonstrate the applicability of the resulting VAE attention: anomaly localization and latent space disentanglement. In anomaly localization, the fact that is used is an abnormal input will result in latent variables that do not conform to the standard Gaussian in gradient backpropagation and visual attention generation. Given an abnormal input to a VAE trained on normal data, attention maps generated using the proposed two methods are presented and how these anomaly attention maps are then used as cues to generate pixel-level binary anomaly masks, helping anomalous images classification and anomaly segmentation. In latent space disentanglement, a new technique is presented which utilizes VAE attention from each latent dimension to enforce new attention disentanglement learning constraints that results in improved attention separability as well as disentanglement performance in training VAE. Experiments are conducted on a wide range of datasets and experimental results validated the effectiveness of the proposed method in visually interpreting variational autoencoders and improving performance of VAEs applications. This chap-

ter has taken the first step to visually interpret VAEs directly from the learned latent space, which freed the gradients-based attention generation from requiring classification/categorization streams in the models. Future work will include studying visual interpretations on a wider range of generative models, for instance, generative adversarial networks (GANs), and these interpretations can potentially boost a variety of down-streaming applications of GANs, e.g., image synthesis, video predictions, etc. In addition, visual interpretations of VAEs can potentially be utilized to design new learning constraints for hybrid deep models which consists of generative modules and discriminative in an end-to-end training for improved performance. Visual attention maps of VAEs bridged the gap of unsupervised learning models' interpretations, and this leveraged broad applications of unsupervised learning with generative models, in particular, in medical procedures, industrial process with limited and scare data.

Chapter 9

Conclusions

This thesis proposed several novel applications for sports analytics, image and language pre-training, face synthesis, scene perceptions, human body mesh recovery and visual interpretations of generative models. In the second chapter, we presented the dataset of highlight videos of soccer players, including two coarse-grained action types of soccer players and six fine-grained actions of players. Detailed annotations are provided for the collected dataset, in terms of action classes, bounding boxes, segmentation maps, and body keypoints of soccer players, and positions of a soccer ball in a game. We leveraged the understanding of complex highlight videos by proposing an energy-motion features aggregation network-***EMA-Net*** to fully exploit energy-based representation of soccer players movements in video sequences and explicit motion dynamics of soccer players in videos for soccer players' fine-grained action analysis. Experimental results and ablation studies validated the proposed approach in recognizing soccer players actions using the collected soccer highlight video datasets.

In the third chapter, we presented RECLIP (Resource-efficient CLIP), a simple method that minimizes computational resource footprint for CLIP (Contrastive Language Image Pretraining). Our approach significantly reduced the training resource requirements both in theory and in practice. Using the same batch size and training epoch, RECLIP achieved highly competitive zero-shot classification and image-text retrieval accuracy with 6 to 8$\times$ less computational resources and 7 to 9$\times$ fewer FLOPs than the baseline. Compared to the state-of-the-art contrastive learning methods, RECLIP demonstrated 5 to 59$\times$ training resource savings while maintaining highly competitive zero-shot classification and retrieval performance. Finally, RECLIP matched the state of the art in transfer learning to open-vocabulary detection tasks, achieving 32 AP$_r$ on LVIS.

In the fourth chapter, we presented a method to visually interpret conditional GANs for facial attribute translation by using a gradient-based attention mechanism. Next, a key innovation was proposed to design new learning objectives for knowledge distillation using attention in generative adversarial training, which resulted in improved synthesized face results, reduced visual confusions and boosted training for GANs in a positive way. Firstly, visual attentions were calculated to provide interpretations for GANs. Secondly, gradient-based visual attentions were used as knowledge to be distilled in a teacher-student paradigm for face synthesis with focus on facial attributes translation tasks in order to improve the performance of the model. Finally, it was shown how "pseudo"-attentions knowledge distillation can be employed during the training of face synthesis networks when teacher and student networks were trained to generate different facial attributes. The proposed approach was validated on facial attribute translation and human expression synthesis with both qualitative and quantitative results being presented.

In the fifth chapter, we proposed a novel framework for improving the performance of self-supervised monocular depth estimation for indoor environments. First, a depth factorization module with transformer-based scale regression network was proposed to estimate a global depth scale factor explicitly, and the predicted scale factor can indicate the maximum depth values. Second, rather than using a single-stage pose estimation strategy as in previous methods, we proposed to utilize a residual pose estimation module to estimate relative camera poses across consecutive frames iteratively. Third, to incorporate extensive coordinates guidance for our residual pose estimation module, we proposed to perform coordinate convolutional encoding directly over the inputs to pose networks. The proposed method was validated on a variety of benchmark indoor datasets, *i.e.*, EuRoC MAV, NYUv2, ScanNet and 7-Scenes, demonstrating the state-of-the-art performance.

In the sixth chapter, we presented a new method for video mesh recovery that divides the human mesh into several local parts following the standard skeletal model. We then modeled the dynamics of each local part with separate recurrent models, with each model conditioned appropriately based on the known kinematic structure of the human body. This resulted in a structure-informed local recurrent learning architecture that can be trained in an end-to-end fashion with available annotations. We conducted a variety of experiments on standard video mesh recovery benchmark datasets such as Human3.6M, MPI-INF-3DHP, and 3DPW, demonstrating the efficacy of our design of modeling local dynamics as well as establishing state-of-the-art results based on standard evaluation metrics.

In the seventh chapter, we presented a method to compute *VAE attention* as a means for interpreting the latent space learned by a VAE. We first presented methods to generate visual attention maps from the learned latent space, and then showed how they can be used in a variety of

applications: localizing anomalies in images, including medical imagery, and improved latent space disentanglement. We conducted extensive experiments on a wide-variety of benchmark datasets to demonstrate the efficacy of the proposed VAE attention.

Bibliography

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *CVPR*, 2019.

[2] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE T-PAMI*, 40:2897–2905, 2016.

[3] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing GANs using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019.

[4] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2018.

[5] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *IJCNN*, 2019.

[6] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*. 2018.

[7] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based SLAM. *CoRR*, 2016.

[8] Kasun Amarasinghe, Kevin Kenney, and Milos Manic. Toward explainable deep neural network based anomaly detection. In *HSI*, 2018.

[9] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

[10] Adria Arbues-Sanguesa, Adrian Martin, Javier Fernandez, Coloma Ballester, and Gloria Haro. Using player's body-orientation to model pass feasibility in soccer. In *CVPR Workshops*, 2020.

[11] Giuseppe Argenziano and H Peter Soyer. Dermoscopy of pigmented skin lesions–a valuable tool for early. *The lancet oncology*, 2(7):443–449, 2001.

[12] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[13] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*, 2019.

[14] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[15] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *The 20th International Conference on Artificial Neural Networks*, 2010.

[16] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

[17] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *ICLR*, 2022.

[18] Sylvio Barbon, Allan Pinto, João Barroso, Fabio Caetano, Felipe Moura, Sergio Cunha, and Ricardo Torres. Sport action mining: Dribbling recognition in soccer. *Multimedia Tools and Applications*, 81(3):4341–4364, 2021.

[19] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI brainlesion workshop*, pages 161–169. Springer, 2018.

[20] Sermetcan Baysal and Pınar Duygulu. Sentioscope: A soccer player tracking system using model field particles. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1350–1362, 2016.

[21] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *ICCV*, 2019.

[22] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

[23] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019.

[24] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

[25] Matt Berseth. Isic 2017-skin lesion analysis towards melanoma detection. *arXiv preprint arXiv:1703.00523*, 2017.

[26] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[27] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k, 2022.

[28] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021.

[29] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019.

[30] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian Reid. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. *arXiv preprint arXiv:2006.02708*, 2020.

[31] Jiawang Bian, Huangying Zhan, Naiyan Wang, TatJin Chin, Chunhua Shen, and Ian Reid. Auto-rectify network for unsupervised indoor depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9802–9813, 2022.

[32] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[33] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[34] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[35] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424, 2018.

[36] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[37] Titus J Brinker, Achim Hekler, Alexander H Enk, Joachim Klode, Axel Hauschild, Carola Berking, Bastian Schilling, Sebastian Haferkamp, Dirk Schadendorf, Tim Holland-Letz, et al. Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. *European Journal of Cancer*, 113:47–54, 2019.

[38] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[39] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[40] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[41] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. 2018.

[42] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *IJRR*, 35(10):1157–1163, 2016.

[43] Fabrizio Caccavale, Ciro Natale, Bruno Siciliano, and Luigi Villani. Six-dof impedance control based on angle/axis representations. *IEEE Transactions on Robotics and Automation*, 15(2):289–300, 1999.

[44] Zixi Cai, Helmut Neher, Kanav Vats, David A. Clausi, and John Zelek. Temporal hockey action recognition via pose and optical flows. In *CVPR Workshops*, 2019.

[45] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2018.

[46] Yuanzhouhan Cao, Tianqi Zhao, Ke Xian, Chunhua Shen, Zhiguo Cao, and Shugong Xu. Monocular depth estimation with augmented ordinal depth relationships. *IEEE Transactions on Image Processing*, 30(8):2674–2682, 2020.

[47] Z Cao, T Simon, SE Wei, YA Sheikh, et al. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[48] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[49] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2016.

[50] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[51] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[52] M Emre Celebi, Noel Codella, and Allan Halpern. Dermoscopy image analysis: overview and future directions. *BHI*, 23(2):474–478, 2019.

[53] Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NeurIPS*, 2016.

[54] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[55] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[56] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018.

[57] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[58] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[59] Eric Z Chen, Xu Dong, Xiaoxiao Li, Hongda Jiang, Ruichen Rong, and Junyan Wu. Lesion attributes segmentation for melanoma detection with multi-task u-net. In *ISBI*, pages 485–488, 2019.

[60] Fan Chen and Christophe De Vleeschouwer. Formulating team-sport video summarization as a resource allocation problem. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(2):193–205, 2011.

[61] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NeurIPS*, 2017.

[62] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *NeurIPS*, 2018.

[63] Shu Chen, Zhengdong Pu, Xiang Fan, and Beiji Zou. Fixing defect of photometric loss for self-supervised monocular depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):1328–1338, 2022.

[64] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*. 2018.

[65] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[66] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, and Denny Zhou. A simple single-scale vision transformer for object localization and instance segmentation, 2022.

[67] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*. 2016.

[68] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *ICLR*, 2017.

[69] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[70] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[71] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015.

[72] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[73] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless Telecommunications Symposium (WTS)*, pages 1–5. IEEE, 2018.

[74] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Abnormal crowd behavior detection and localization using maximum sub-sequence search. In *Proceedings of the 4th ACM/IEEE international workshop on Analysis and retrieval of tracked events and motion in imagery stream*, pages 49–58. ACM, 2013.

[75] William Ching, John Robinson, and Mark McEntee. Patient-based radiographic exposure factor selection: a systematic review. *Journal of medical radiation sciences*, 61(3):176–190, 2014.

[76] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[77] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[78] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Starganv2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

[79] Anthony Cioppa, Adrien Deliege, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-v3: Scaling up soccernet with multi-view spatial localization and re-identification. *Sci Data*, June 2022.

[80] Anthony Cioppa, Adrien Deliege, Floriane Magera, Silvio Giancola, Olivier Barnich, Bernard Ghanem, and Marc Van Droogenbroeck. Camera calibration and player localization in soccernet-v2 and investigation of their representations for action spotting. In *CVPR Workshops*, 2021.

[81] Anthony Cioppa, Adrien Deliege, and Marc Van Droogenbroeck. A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games. In *CVPR Workshops*, 2018.

[82] Anthony Cioppa, Silvio Giancola, Adrien Deliège, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In *CVPR Workshops*, 2022.

[83] Lei Clifton, David A Clifton, Peter J Watkinson, and Lionel Tarassenko. Identification of patient deterioration in vital-sign data using one-class support vector machines. In *FedCSIS*, 2011.

[84] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[85] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi). In *ISBI*, 2018.

[86] Noel CF Codella, Q-B Nguyen, Sharath Pankanti, David A Gutman, Brian Helba, Allan C Halpern, and John R Smith. Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM Journal of Research and Development*, 61(4/5):5–1, 2017.

[87] Taco Cohen and Max Welling. Learning the irreducible representations of commutative lie groups. In *ICML*, 2014.

[88] Taco Cohen and Max Welling. Transformation properties of learned visual representations. *CoRR*, abs/1412.7659, 2014.

[89] Marc Combalia, Noel CF Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C Halpern, Susana Puig, et al. Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*, 2019.

[90] Marc Combalia, Ferran Hueto, Susana Puig, Josep Malvehy, and Veronica Vilaplana. Uncertainty estimation in deep neural networks for dermoscopic image classification. In *CVPR Workshops*, 2020.

[91] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[92] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv preprint arXiv:2002.03734*, 2020.

[93] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *CVPR Workshops*, 2021.

[94] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[95] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[96] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[97] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[98] saurabh desai and Harish Guruprasad Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*, 2020.

[99] Guillaume Desjardins, Aaron C. Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *ArXiv*, abs/1210.5474, 2012.

[100] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019.

[101] Xing Di and Vishal M. Patel. Facial synthesis from visual attributes via sketch using multiscale generators. *IEEE Transactions on Biometrics, Identity and Behavior*, 2(1):55–67, 2020.

[102] Xing Di and Vishal M. Patel. Multimodal face synthesis from visual attributes. *IEEE Transactions on Biometrics, Identity and Behavior*, 3(3):427–439, 2021.

[103] Xing Di, Benjamin S. Riggan, Shuowen Hu, Nathaniel J. Short, and Vishal M. Patel. Multiscale thermal to visible face verification via attribute guided synthesis. *IEEE Transactions on Biometrics, Identity and Behavior*, 3(2):266–280, 2021.

[104] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019.

[105] Santosh K Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[106] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. In *NeurIPS*, 2019.

[107] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *CVPR*, 2010.

[108] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Shuyang Gu, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vitb and vit-l on imagenet, 2022.

[109] Xiaoyi Dong, Yinglin Zheng, Jianmin Bao, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining, 2022.

[110] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[111] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, 2016.

[112] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *CVPR*, 2022.

[113] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *ICCV*, 2003.

[114] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[115] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[116] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured disentangled representations. In *AISTATS*, 2019.

[117] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[118] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[119] Babak Fakhar, Hamidreza Rashidy Kanan, and Alireza Behrad. Event detection in soccer videos using unsupervised learning of spatio-temporal features based on pooled spatial pyramid model. *Multimedia Tools Applications*, 78(12):16995–17025, 2019.

[120] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI*, 2018.

[121] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *WACV*, 2020.

[122] Mehrnaz Fani, Helmut Neher, David A. Clausi, Alexander Wong, and John Zelek. Hockey action recognition via integrated stacked hourglass network. In *CVPR Workshops*, 2017.

[123] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020.

[124] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[125] Tomaso Fontanini, Eleonora Iotti, Luca Donati, and Andrea Prati. MetalGAN: Multi-domain label-less image synthesis using cGANs and meta-learning. *Neural Networks*, 131:185–200, 2020.

[126] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[127] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[128] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, 2019.

[129] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, 2019.

[130] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[131] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[132] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[133] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecka, and Ziyan Wu. Hierarchical kinematic human mesh recovery. In *ECCV*, 2020.

[134] Sebastian Gerke, Karsten Muller, and Ralf Schafer. Soccer jersey number recognition using convolutional neural networks. In *ICCV Workshops*, 2015.

[135] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. *MethodsX*, 7:100864, 2020.

[136] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *CVPR Workshops*, 2018.

[137] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. In *CVPR Workshops*, 2021.

[138] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[139] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[140] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[141] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.

[142] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press.

[143] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[144] Ian Goodfellow, Honglak Lee, Quoc Le, Andrew Saxe, and Andrew Ng. Measuring invariances in deep networks. *NeurIPS*, 2009.

[145] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[146] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *CVPR*, 2019.

[147] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Chengzhou Tang, Siyu Zhu, and Ping Tan. Dro: Deep recurrent optimizer for structure-from-motion. *arXiv preprint arXiv:2103.13201*, 2021.

[148] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.

[149] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020.

[150] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017.

[151] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Yunhe Wang, and Chang Xu. Fastmim: Expediting masked image modeling pre-training for vision, 2022.

[152] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018.

[153] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[154] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *CVPR*, 2019.

[155] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[156] Holger A Haenssle, Christine Fink, Roland Schneiderbauer, Ferdinand Toberer, Timo Buhl, Andreas Blum, A Kalloo, A Ben Hadj Hassen, Luc Thomas, A Enk, et al. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Annals of Oncology*, 29(8):1836–1842, 2018.

[157] J. Han and Bir Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):316–322, 2006.

[158] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018.

[159] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[160] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[161] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[162] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[163] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[164] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019.

[165] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning, Springer, 2018.

[166] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

[167] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[168] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[169] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[170] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[171] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[172] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[173] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, pages 654–661. IEEE, 2005.

[174] Yutaro Honda, Rei Kawakami, Ryota Yoshihashi, Kenta Kato, and Takeshi Naemura. Pass receiver prediction in soccer using video and players' trajectories. In *CVPR Workshops*, 2022.

[175] Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *arXiv:2210.07224*, 2022.

[176] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[177] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[178] Xun Huang, Yixuan Li, Omid Poursaeed, John E. Hopcroft, and Serge J. Belongie. Stacked generative adversarial networks. *CVPR*, 2016.

[179] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[180] Zhenhua Huang, Shunzhi Yang, MengChu Zhou, Zhetao Li, Zheng Gong, and Yunwen Chen. Feature map distillation of thin nets for low-resolution object recognition. *IEEE Transactions on Image Processing*, 31:1364–1379, 2022.

[181] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[182] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013.

[183] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *CVPR*, 2020.

[184] Hossam Isack, Christian Haene, Cem Keskin, Sofien Bouaziz, Yuri Boykov, Shahram Izadi, and Sameh Khamis. Repose: Learning deep kinematic priors for fast human pose estimation. *arXiv preprint arXiv:2002.03933*, 2020.

[185] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[186] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015.

[187] Pan Ji, Runze Li, Bir Bhanu, and Yi Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *ICCV*, 2021.

[188] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[189] H. Jiang, Y. Lu, and J. Xue. Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In *ICTAI*, 2016.

[190] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021.

[191] Dakai Jin, Dazhou Guo, Tsung-Ying Ho, Adam P. Harrison, Jing Xiao, Chen-Kan Tseng, and Le Lu. Accurate esophageal gross tumor volume segmentation in pet/ct using two-stream chained 3d deep network fusion. In *MICCAI*, 2019.

[192] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

[193] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[194] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *CVPR*, 2019.

[195] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Label-noise robust generative adversarial networks. In *CVPR*, 2019.

[196] Srikrishna Karanam, Ren Li, Fan Yang, Wei Hu, Terrence Chen, and Ziyan Wu. Towards contactless patient positioning. *IEEE Transactions on Medical Imaging*, 39(8):2701–2710, 2020.

[197] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[198] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[199] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[200] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of styleGAN. In *CVPR*, 2020.

[201] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[202] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[203] Hirokatsu Kataoka, Tenga Wakamiya, Kensho Hara, and Yutaka Satoh. Would mega-scale datasets further enhance spatiotemporal 3d cnns? *CoRR*, 2020.

[204] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017.

[205] Daejin Kim, Mohammad Azam Khan, and Jaegul Choo. Not just compete, but collaborate: Local image-to-image translation via cooperative mask prediction. In *CVPR*, 2021.

[206] Daejin Kim, Mohammad Azam Khan, and Jaegul Choo. Not just compete, but collaborate: Local image-to-image translation via cooperative mask prediction. In *CVPR*, 2021.

[207] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, 2023.

[208] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

[209] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

[210] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[211] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR*, 2015.

[212] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[213] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[214] Harold Kittler, H Pehamberger, K Wolff, and MJTIO Binder. Diagnostic accuracy of dermoscopy. *The Lancet Oncology*, 3(3):159–165, 2002.

[215] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[216] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[217] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019.

[218] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, June 2019.

[219] Longteng Kong, Duoxuan Pei, Rui He, Di Huang, and Yunhong Wang. Spatio-temporal player relation modeling for tactic recognition in sports videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):6086–6099, 2022.

[220] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[221] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. 2012.

[222] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[223] Kaustubh Milind Kulkarni and Sucheth Shenoy. Table tennis stroke recognition using two-dimensional human pose estimation. In *CVPR Workshops*, 2021.

[224] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Backpropagated gradient representations for anomaly detection. In *ECCV*, 2020.

[225] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

[226] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *CVPR*, 2014.

[227] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[228] David Lange. High school athletics participation survey 2018-19.

[229] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010.

[230] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[231] Jie Lei, Xinlei Chen, Ning Zhang, Mengjiao Wang, Mohit Bansal, Tamara L. Berg, and Licheng Yu. Loopitr: Combining dual and cross encoder architectures for image-text retrieval, 2022.

[232] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015.

[233] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, 2019.

[234] Gen Li, Shikun Xu, Xiang Liu, Lei Li, and Changhu Wang. Jersey number recognition with semi-supervised spatial transformer network. In *CVPR Workshops*, June 2018.

[235] Jianhai Li, Unni K Udayasankar, Thomas L Toth, John Seamans, William C Small, and Mannudeep K Kalra. Automatic patient centering for MDCT: effect on radiation dose. *American journal of roentgenology*, 188(2):547–552, 2007.

[236] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *ICCV*, 2017.

[237] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018.

[238] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018.

[239] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *T-PAMI*, 42(12):2996–3010, 2019.

[240] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Guided attention inference network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(12):2996–3010, 2020.

[241] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional GANs. In *CVPR*, 2020.

[242] Runze Li and Bir Bhanu. Fine-grained visual dribbling style analysis for soccer videos with augmented dribble energy image. In *CVPR Workshops*, 2019.

[243] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *CVPR*, 2020.

[244] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Knowledge distillation from few samples. 2018.

[245] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *T-PAMI*, 36(1):18–32, 2013.

[246] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022.

[247] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. *preprint :2212.00794*, 2022.

[248] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[249] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[250] Zhenyu Li, Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Superpixel masking and inpainting for self-supervised anomaly detection. In *BMVC*, 2020.

[251] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[252] Xiaoyu Zhao Chenkai Zeng Yichun Yang Limin Wang, Yutao Cui. Sportsmot: A large-scale multi-object tracking dataset in sports scenes.

[253] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, pages 1954–1963, June 2021.

[254] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[255] Yu Lin, Yigong Wang, Yifan Li, Yang Gao, Zhuoyi Wang, and Latifur Khan. Attention-based spatial guidance for image-to-image translation. In *WACV*, 2021.

[256] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015.

[257] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2015.

[258] Haojie Liu, Ming Lu, Zhan Ma, Fan Wang, Zhihuang Xie, Xun Cao, and Yao Wang. Neural video coding using multiscale motion compensation and spatiotemporal context model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):3182–3196, 2021.

[259] Hengyue Liu and Bir Bhanu. Pose-guided R-CNN for jersey number recognition in sports. In *CVPR Workshops*, 2019.

[260] Hengyue Liu and Bir Bhanu. Jede: Universal jersey number detector for sports. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7894–7909, 2022.

[261] Jiachen Liu, Pan Ji, Nitin Bansal, Changjiang Cai, Qingan Yan, Xiaolei Huang, and Yi Xu. Planemvs: 3d plane reconstruction from multi-view stereo. In *CVPR*, 2022.

[262] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[263] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*, 2019.

[264] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017.

[265] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019.

[266] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018.

[267] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, 2015.

[268] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

[269] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *CVPR*, 2020.

[270] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J. Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *CVPR*, 2020.

[271] Xingyu Liu, , Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. 2019.

[272] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

[273] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

[274] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[275] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[276] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022.

[277] Zhi-Song Liu, Wan-Chi Siu, and Yui-Lam Chan. Photo-realistic image super-resolution via variational autoencoders. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4):1351–1365, 2021.

[278] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[279] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. *NeurIPS*, 2019.

[280] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015.

[281] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, October 2015.

[282] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *CVPR*, 2019.

[283] Keyu Lu, Jianhui Chen, James J. Little, and Hangen He. Light cascaded convolutional neural networks for accurate player detection. In *BMVC*, 2017.

[284] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics*, 39(4):71–1, 2020.

[285] Chunyan Ma, Ji Fan, Jinghao Yao, and Tao Zhang. Npu rgbd dataset and a feature-enhanced lstm-dgcn method for action recognition of basketball players+. *Applied Sciences*, 11(10), 2021.

[286] Furong Ma, Guiyu Xia, and Qingshan Liu. Spatial consistency constrained gan for human motion transfer. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):730–742, 2022.

[287] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NIPS*. 2017.

[288] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

[289] Zubair Martin, Sharief Hendricks, and Amir Patel. Automated tackle injury risk assessment in contact-based sports - a rugby union example. In *CVPR Workshops*, 2021.

[290] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

[291] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Real-time convolutional networks for depth-based human pose estimation. In *IROS*, 2018.

[292] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *ICML*, 2019.

[293] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[294] Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. In *CVPR*, 2019.

[295] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.

[296] Kaustubha Mendhurwar, Gaurav Handa, Leixiao Zhu, Sudhir Mudur, Etienne Beauchesne, Marc LeVangie, Aiden Hallihan, Abbas Javadtalab, and Tiberiu Popa. A system for acquisition and modelling of ice-hockey stick shape deformation from player shot videos. In *CVPR Workshops*, 2020.

[297] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[298] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[299] Tomoyuki Mukasa, Jiu Xu, and Bjorn Stenger. 3d scene mesh from cnn depth predictions and sparse monocular slam. In *ICCV Workshops*, pages 921–928, 2017.

[300] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[301] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18(1):209, 2018.

[302] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *CoRR*, 2021.

[303] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[304] Naoki Nonaka, Ryo Fujihira, Monami Nishio, Hidetaka Murakami, Takuya Tajima, Mutsuo Yamada, Akira Maeda, and Jun Seita. End-to-end high-risk tackle detection system for rugby. In *CVPR Workshops*, 2022.

[305] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015.

[306] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans, 2017.

[307] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, 2018.

[308] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

[309] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[310] Yangjun Ou, Zhenzhong Chen, and Feng Wu. Multimodal local-global attention network for affective video content analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1901–1914, 2021.

[311] Andre GC Pacheco and Renato A Krohling. Recent advances in deep learning applied to skin cancer detection. *arXiv preprint arXiv:1912.03280*, 2019.

[312] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.

[313] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

[314] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshops*, 2017.

[315] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[316] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, 2018.

[317] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, 2018.

[318] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*, 2019.

[319] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *CoRR*, abs/2111.10050, 2021.

[320] Hemanth Pidaparthy, Michael H. Dowling, and James H. Elder. Automatic play segmentation of hockey videos. In *CVPR Workshops*, 2021.

[321] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*, 2018.

[322] A. J. Piergiovanni and Michael S. Ryoo. Early detection of injuries in MLB pitchers from video. *CoRR*, 2019.

[323] AJ Piergiovanni and Michael S. Ryoo. Fine-grained activity recognition in baseball videos. In *CVPR Workshops*, 2018.

[324] AJ Piergiovanni and Michael S. Ryoo. Early detection of injuries in mlb pitchers from video. In *CVPR Workshops*, 2019.

[325] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *CVIU*, 208-209:103219, 05 2021.

[326] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *ICPR Workshops and Challenges*, 2021.

[327] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[328] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

[329] PyTorch. Keypoint r-cnn.

[330] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018.

[331] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017.

[332] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[333] Mengshi Qi, Yunhong Wang, Annan Li, and Jiebo Luo. Sports video captioning via attentive motion representation and group relationship modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2617–2633, 2020.

[334] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[335] Julian Quiroga, Henry Carrillo, Edisson Maldonado, John Ruiz, and Luis M. Zapata. As seen on tv: Automatic basketball video production using gaussian-based actionness and game states recognition. In *CVPR Workshops*, 2020.

[336] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[337] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[338] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[339] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[340] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2022.

[341] Mahdyar Ravanbakhsh, Enver Sangineto, Moin Nabi, and Nicu Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. In *WACV*, 2019.

[342] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *ICML*, 2019.

[343] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[344] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[345] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[346] Vito Reno, Nicola Mosca, Roberto Marani, Massimiliano Nitti, Tiziana D'Orazio, and Ettore Stella. Convolutional neural networks based ball detection in tennis games. In *CVPR Workshops*, 2018.

[347] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and variational inference in deep latent gaussian models. *ArXiv*, abs/1401.4082, 2014.

[348] Graham Thomas Adrian Hilton Jim Little Michele Merler Rikke Gade, Thomas Moeslund. 8th international workshop on computer vision in sports (cvsports) at cvpr 2022.

[349] Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). In *CVPR*, 2019.

[350] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[351] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[352] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018.

[353] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[354] Ryan Sanford, Siavash Gorji, Luiz G. Hafemann, Bahareh Pourbabaee, and Mehrsan Javan. Group activity detection from trajectory and video data in soccer. In *CVPR Workshops*, 2020.

[355] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017.

[356] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NeurIPS*, 2018.

[357] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020.

[358] Saikat Sarkar, Amlan Chakrabarti, and Dipti Prasad Mukherjee. Generation of ball possession statistics in soccer using minimum-cost flow network. In *CVPR Workshops*, 2019.

[359] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In *NeurIPS*, 2006.

[360] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

[361] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *IPMI*, 2017.

[362] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Comput.*, 4(6):863–879, November 1992.

[363] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, 1999.

[364] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[365] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[366] Johannes L. Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[367] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, Tosmanov, Dmitry Kruchinin, Artyom Zankevich, Dmitriy Sidnev, Maksim Markelov, Johannes, Mathis Chenuet, A-Andre, Telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, Vugia Truong, Zliang7, Lizhming, and Tritin Truong. opencv/cvat: v1.1.0, August 2020.

[368] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[369] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[370] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[371] Jialie Shen, Dacheng Tao, and Xuelong Li. Modality mixture projections for semantic video event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1587–1596, 2008.

[372] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[373] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, 2013.

[374] Md Mahfuzur Rahman Siddiquee, Zongwei Zhou, Nima Tajbakhsh, Ruibin Feng, Michael B Gotway, Yoshua Bengio, and Jianming Liang. Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization. In *ICCV*, 2019.

[375] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2019. *CA: a cancer journal for clinicians*, 69(1):7–34, 2019.

[376] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012.

[377] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[378] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, pages 15638–15650, 2022.

[379] Maneet Singh, Shruti Nagpal, Richa Singh, and Mayank Vatsa. Dual directed capsule network for very low resolution image recognition. In *ICCV*, 2019.

[380] Maneet Singh, Shruti Nagpal, Richa Singh, and Mayank Vatsa. Derivenet for (very) low resolution image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6569–6577, 2022.

[381] Vivek Singh, Kai Ma, Birgi Tamersoy, Yao-Jen Chang, Andreas Wimmer, Thomas O'Donnell, and Terrence Chen. DARWIN: Deformable patient avatar representation with deep image network. In *MICCAI*, 2017.

[382] Christoph Sinz, Philipp Tschandl, Cliff Rosendahl, Bengu Nisa Akay, Giuseppe Argenziano, Andreas Blum, Ralph P Braun, Horacio Cabo, Jean-Yves Gourhant, Juergen Kreusch, et al. Accuracy of dermatoscopy for the diagnosis of nonpigmented cancers of the skin. *Journal of the American Academy of Dermatology*, 77(6):1100–1109, 2017.

[383] Minsoo Song, Seokjae Lim, and Wonjun Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4381–4393, 2021.

[384] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

[385] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[386] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. *CVPR*, 2021.

[387] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J. Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, pages 11179–11188, October 2021.

[388] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, YiLi Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *ICCV*, 2019.

[389] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *CVPR*, 2022.

[390] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. 2018.

[391] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey E. Hinton. Tensor analyzers. In *ICML*, 2013.

[392] Yichuan Tang, Nitish Srivastava, and Ruslan Salakhutdinov. Learning generative models with visual attention. In *NIPS*, 2013.

[393] Mostafa Tavassolipour, Mahmood Karimian, and Shohreh Kasaei. Event detection and summarization in soccer videos using bayesian network and copula. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(2):291–304, 2014.

[394] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2018.

[395] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In *NeurIPS*, 2021.

[396] Rajkumar Theagarajan and Bir Bhanu. An automated system for generating tactical performance statistics for individual soccer players from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2):632–646, 2021.

[397] Rajkumar Theagarajan, Federico Pala, Xiu Zhang, and Bir Bhanu. Soccer: Who has the ball? generating visual analytics and player statistics. In *CVPR Workshops*, 2018.

[398] Fangzheng Tian, Yongbin Gao, Zhijun Fang, Yuming Fang, Jia Gu, Hamido Fujita, and Jenq-Neng Hwang. Depth estimation using a self-supervised network based on cross-layer feature fusion and the quadtree constraint. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4):1751–1766, 2022.

[399] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. In *ECCV*, 2020.

[400] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021.

[401] Max Torop, Sandesh Ghimire, Wenqian Liu, Dana H Brooks, Octavia Camps, Milind Rajadhyaksha, Jennifer Dy, and Kivanc Kose. Unsupervised approaches for out-of-distribution dermoscopic lesion detection. *arXiv preprint arXiv:2111.04807*, 2021.

[402] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

[403] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[404] Philipp Tschandl, Noel Codella, Bengü Nisa Akay, Giuseppe Argenziano, Ralph P Braun, Horacio Cabo, David Gutman, Allan Halpern, Brian Helba, Rainer Hofmann-Wellenhof, et al. Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. *The Lancet Oncology*, 20(7):938–947, 2019.

[405] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):1–9, 2018.

[406] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[407] Takamasa Tsunoda, Yasuhiro Komori, Masakazu Matsugu, and Tatsuya Harada. Football action recognition using hierarchical lstm. In *CVPR Workshops*, 2017.

[408] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[409] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017.

[410] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, pages 5038–5047, 2017.

[411] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *NeurIPS*, 2019.

[412] Bastien Vanderplaetse and Stephane Dupont. Improved soccer action spotting using both audio and video streams. In *CVPR Workshops*, 2020.

[413] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[414] Kanav Vats, Mehrnaz Fani, David A. Clausi, and John Zelek. Puck localization and multi-task event recognition in broadcast hockey videos. In *CVPR Workshops*, 2021.

[415] Kanav Vats, William McNally, Pascale Walters, David A. Clausi, and John S. Zelek. Ice hockey player identification via transformers and weakly supervised learning. In *CVPR Workshops*, 2022.

[416] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018.

[417] Hung Vu, Tu Dinh Nguyen, Trung Le, Wei Luo, and Dinh Phung. Robust anomaly detection in videos using multilevel representations. In *AAAI*, 2019.

[418] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[419] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

[420] Jianqiang Wang, Hao Zhu, Haojie Liu, and Zhan Ma. Lossy point cloud geometry compression via end-to-end learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12):4909–4923, 2021.

[421] Josiah Wang, Katja Markert, Mark Everingham, et al. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009.

[422] Lezi Wang, Ziyan Wu, Srikrishna Karanam, Kuan-Chuan Peng, Rajat Vikram Singh, Bo Liu, and Dimitris N. Metaxas. Sharpen focus: Learning with attention separability and consistency. In *ICCV*, 2019.

[423] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, 2021.

[424] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *CoRR*, abs/2109.08472, 2021.

[425] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[426] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *CVPR*, 2019.

[427] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *ICCV*, 2021.

[428] Shenzhi Wang, Liwei Wu, Lei Cui, and Yujun Shen. Glancing at the patch: Anomaly localization with global and local feature comparison. In *CVPR*, 2021.

[429] Siqi Wang, En Zhu, Jianping Yin, and Fatih Porikli. Video anomaly detection and localization by local motion based joint video representation and ocelm. *Neurocomputing*, 277:161–175, 2018.

[430] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018.

[431] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[432] Zengkai Wang, Junqing Yu, and Yunfeng He. Soccer video event annotation by synchronization of attack–defense clips and match reports with coarse-grained time information. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(5):1104–1117, 2017.

[433] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *WACV*, 2022.

[434] Zhe Wang, Liyan Chen, Shauray Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. In *Arxiv*, 2019.

[435] Zhe Wang, Daeyun Shin, and Charless Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *ECCV Workshops*, 2020.

[436] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision, 2021.

[437] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *CVPR*, 2021.

[438] Floris Weers, Vaishaal Shankar, Angelos Katharopoulos, Yinfei Yang, and Tom Gunter. Self supervision does not help natural language supervision at scale, 2023.

[439] Bill Wilson. Premier league club revenues soar to £4.5bn.

[440] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A multigrid method for efficiently training video models. In *CVPR*, June 2020.

[441] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *CVPR*, 2019.

[442] Lifang Wu, Zhou Yang, Jiaoyu He, Meng Jian, Yaowen Xu, Dezhong Xu, and Chang Wen Chen. Ontology-based global and collective motion patterns for event classification in basketball videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):2178–2190, 2020.

[443] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. RelGAN: Multi-domain image-to-image translation via relative attributes. In *ICCV*, 2019.

[444] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. TextureGAN: Controlling deep image synthesis with texture patches. In *CVPR*, 2018.

[445] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019.

[446] Sitao Xiang, Yuming Gu, Pengda Xiang, Menglei Chai, Hao Li, Yajie Zhao, and Mingming He. Disunknown: Distilling unknown factors for disentanglement learning. In *ICCV*, 2021.

[447] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[448] Chengming Xu, Yanwei Fu, Bing Zhang, Zitian Chen, Yu-Gang Jiang, and Xiangyang Xue. Learning to score figure skating sport videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4578–4590, 2020.

[449] Jianfeng Xu, Lertniphonphan Kanokphan, and Kazuyuki Tasaka. Fast and accurate object detection using image cropping/resizing in multi-view 4k sports videos. In *MM Sports Workshop*, 2018.

[450] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[451] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[452] Lan Yang, Kaiyue Pang, Honggang Zhang, and Yi-Zhe Song. Sketchaa: Abstract representation for abstract sketches. In *ICCV*, 2021.

[453] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *CVPR*, 2021.

[454] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2021.

[455] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, 2019.

[456] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[457] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022.

[458] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng-Ann Heng. Automated melanoma recognition in dermoscopy images via very deep residual networks. *T-MI*, 36(4):994–1004, 2016.

[459] Qien Yu, Muthusubash Kavitha, and Takio Kurita. Extensive framework based on novel convolutional and variational autoencoder based on maximization of mutual information for anomaly detection. *Neural Computing and Applications*, pages 1–23, 2021.

[460] Zehao Yu, Lei Jin, and Shenghua Gao. P^2net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *ECCV*, 2020.

[461] Zhen Yu, Xudong Jiang, Feng Zhou, Jing Qin, Dong Ni, Siping Chen, Baiying Lei, and Tianfu Wang. Melanoma recognition in dermoscopy images via aggregated deep convolutional features. *T-BE*, 66(4):1006–1016, 2018.

[462] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[463] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[464] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[465] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[466] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[467] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[468] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022.

[469] Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? In *ICRA*, 2020.

[470] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *CVPR*, 2020.

[471] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[472] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[473] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[474] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1947–1962, 2018.

[475] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *ICCV*, 2019.

[476] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[477] Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *IJCV*, 126:1084–1102, 2016.

[478] Ke Zhang, Yukun Su, Xiwang Guo, Liang Qi, and Zhenbing Zhao. Mu-gan: Facial attribute editing based on multi-attention mechanism. *IEEE/CAA Journal of Automatica Sinica*, 8(9):1614–1626, 2020.

[479] Ke Zhang, Yukun Su, Xiwang Guo, Liang Qi, and Zhenbing Zhao. MU-GAN: facial attribute editing based on multi-attention mechanism. *CoRR*, 2020.

[480] Sijia Zhang, Maoguo Gong, Yu Xie, A. K. Qin, Hao Li, Yuan Gao, and Yew-Soon Ong. Influence-aware attention networks for anomaly detection in surveillance videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8):5427–5437, 2022.

[481] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, December 2013.

[482] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *ArXiv*, abs/1706.02262, 2017.

[483] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020.

[484] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *ACM MM*, 2017.

[485] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. Re-identification with consistent attentive siamese networks. In *CVPR*, 2019.

[486] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *CVPR*, 2019.

[487] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *CVPR*, 2019.

[488] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[489] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[490] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[491] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *KDD*, 2017.

[492] Fei Zhou, Lei Zhang, and Wei Wei. Meta-generating deep attentive metric for few-shot classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):6863–6873, 2022.

[493] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *ECCV*, 2018.

[494] Joey Tianyi Zhou, Le Zhang, Zhiwen Fang, Jiawei Du, Xi Peng, and Yang Xiao. Attention-driven loss for anomaly detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4639–4647, 2020.

[495] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *ICCV*, pages 8618–8627, 2019.

[496] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022.

[497] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, Zaiwang Gu, Jiang Liu, and Shenghua Gao. Encoding structure-texture relation with p-net for anomaly detection in retinal images. In *ECCV*, 2020.

[498] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[499] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *ECCV*, 2016.

[500] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[501] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018.

[502] Yuliang Zou, Pan Ji, , Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. In *ECCV*, 2020.

[503] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.

[504] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.

[505] Yiming Zuo, Weichao Qiu, Lingxi Xie, Fangwei Zhong, Yizhou Wang, and Alan L. Yuille. Craves: Controlling robotic arm with a vision-based economic system. In *CVPR*, 2019.