UNIVERSITY OF CALIFORNIA

Los Angeles

Learning Fluents for Task Representation

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Yang Liu

2019

ABSTRACT OF THE DISSERTATION

Learning Fluents for Task Representation

by

Yang Liu
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2019
Professor Song-Chun Zhu, Chair

This dissertation focuses on a crucial challenge which hasn't received enough attention in past years - fluents change. Fluents are time-varying attributes of an entity or a group of entities. Fluents describe the state change of humans, objects and environment, which provides the interpretable representation of them in video analysis and task planing, while popular method such as 3D convolution descriptor lacks such property and remains a black box to people. This dissertation is mainly divided into two parts. In the first part, we are discussing the nature of fluents, which can be further divided into appearance, geometry and topology. We have developed a generative model with encoder-decoder mechanics to map the fluents between image space and latent space. To disentangle fluents like appearance and geometry, we have designed different encoder-decoder networks. With this kind of design, the object in image will be mapped to appearance and geometric fluents vector separately. Moreover, in the geometric fluents space, we are trying to learn the intuitive physics with the synthesis datasets which includes object interactions like collision and gravity. In the second part, we have proposed a framework to represent the task from the perspective of fluents change, which is different from traditional approach. Task is a series of actions to finish certain goal, which can be represented by a group of fluents change. We have collected datasets in both real scene and VR scene, the experiments on both datasets demonstrate the strength of our methods.

The dissertation of Yang Liu is approved.

Ying Nian Wu

Demetri Terzopoulos

Tao Gao

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2019

To my family,

who supports me all the time along the way

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First I want to show my deepest gratitude to my advisor, Prof. Song-Chun Zhu, for his helpful instructions in this five-year journey. Prof. Zhu is truly an erudite, sharp-eyed and farseeing scholar for his unique insight in not only computer vision but also artificial intelligence, cognition and psychology. I always feel fortunate and grateful to study and do research in VCLA, which gives me the opportunity to study many interesting and fundamental problems and help me to think deeply in the future life. I would also like to thank my committee members: Prof. Tao Gao, Prof. Hongjing Lu, Prof. Demetri Terzopoulos and Prof. Ying Nian Wu for their helpful instructions.

Second, I would like to thank the members in VCLA. In this five years, we have built a close friendship with each other. I really enjoy the time when we were discussing, collaborating and having fun. Particularly, I would like to thank Prof. Ping Wei, Prof. Xiaobai Liu, Prof. Xianglei Xing, Dr. Zhixiong Nan, Ruiqi Gao, Dr. Yuanlu Xu, Dr. Tianmin Shu, Ran Gong, Dr. Jianwen Xie who work close with me in various projects. Many thanks to my other friends and colleagues in VCLA for their supports: Dr. Bo Li, Dr. Keze Wang, Dr. Yixin Zhu, Dr. Dan Xie, Dr. Joey Yu, Dr. Yang Lu, Yixin Chen, Lifeng Fan, Xiaofeng Gao, Siyuan Huang, Qing Li, Siyuan Qi, Shu Wang, Tao Yuan.

Last, I am really feeling grateful to my parents who took care of and supported me in the past years and my wife Chenxi for her love.

VITA

2014–2019	Graduate Research Assistant, UCLA VCLA, Dr. Song-chun Zhu
2018	Intern, Amazon Go
2016	Ph.D Candidate in Statistics, UCLA,
2010–2014	B.S. (Automation), Tsinghua University, China

PUBLICATIONS

Recognizing Unseen Attribute-Object Pair with Generative Model. Zhixiong Nan*, Yang Liu*, Nanning Zheng, Song-Chun Zhu. *AAAI*, 2019

Where and Why Are They Looking? Jointly Inferring Human Attention and Intentions in Complex Tasks. Ping Wei, Yang Liu, Tianmin Shu, Nanning Zheng, Song-Chun Zhu . *CVPR*, 2018

Jointly Recognizing Object Fluents and Tasks in Egocentric Videos. Yang Liu*, Ping Wei*, Song-Chun Zhu. *ICCV*, 2017

Composing Hierarchical Structures for Multi-view Multi-object Tracking. Yuanlu Xu, Xiaobai Liu, Yang Liu, Song-Chun Zhu . *CVPR*, 2016

CHAPTER 1

Introduction

In past years, task recognition and planning have been studied intensively. Early works [SLC04, WKS11] use handcrafted features to capture spatial and temporal information. With the development of deep neural networks, some deep learned features [SZ14a, FPW16, FPW17] have been proposed with improved accuracy. Another trend is to characterize the task as an ST-AOG [WZZ13a, WZZ17] with each node representing the component of the task.

However, little attention has been received on the fluent change of objects during the task. Task is a set of actions to finish one or a series of goals. The goal of the task can be represented by the fluents change of the task-related objects. A fluent is a time-varying attribute of an entity or a group of entities, and its values are the specific states of the attribute [FZ15, Mue15].

Fluents represent the object dynamics in an image pair or images sequence, which provides a new perspective to some traditional computer vision problems. Based on the nature of fluents, we can group them into two categories: self fluents and mutual fluents. Self fluents describe the variable nature of the entity group of entities itself while mutual fluents describe the relationship between entities.

1.1 Self Fluents

From the perspective of computer vision, self fluents can be grouped into visible fluents which we can perceive directly from images such as brightness and position and invisible fluents like temperature, pressure or emotion of the humans. For invisible fluents, we treat them as hidden state of the entity, which we may be able to infer from the visual information. For example, when we observe that the water has been heated for 10 minutes or the bubbles begin to rise from the bottom, we can infer that the water should be ‘hot’.

Furtherer, visible fluents can be disentangled as appearance, geometry and topology according to its nature which could act as actionable features for modeling the human activities in vision community. Figure 1.1 illustrates the changes of these three types.

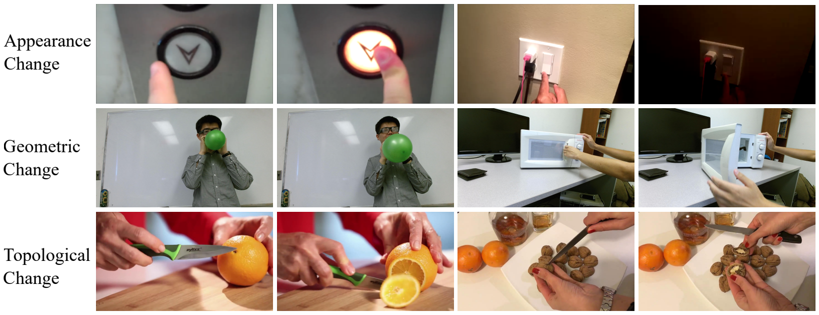


Figure 1.1: Illustration of three types of fluent changes. Which are change in color, change in brightness, scaling, rotation, into slices, into two parts respectively.

- **Appearance change** is defined as the value change of pixels on object exterior. Based on different change types, it can be divided into change in color, brightness and texture. Pressing the button changes the color of button, and turning on the light changes the brightness of the environment.

- **Geometric change** includes objects’ change in position, orientation and shape. Change of position and orientation are also called rigid body motion in terms of physics, including translation and rotation. The shape change includes affine transformation like scaling and shear and non-affine transformation such as torsion and bending. Note that all these changes don’t change the topological structure of objects.

- **Topological change** is defined on a group of entities. Topology is a corpus of multiple entities and their connections as shown in Figure 1.2. Entities belong to the same group share the self fluent of the group like the velocity. The topological change will add or break some connections to change the topological structure of the group such as cutting the apple, peeling the banana.

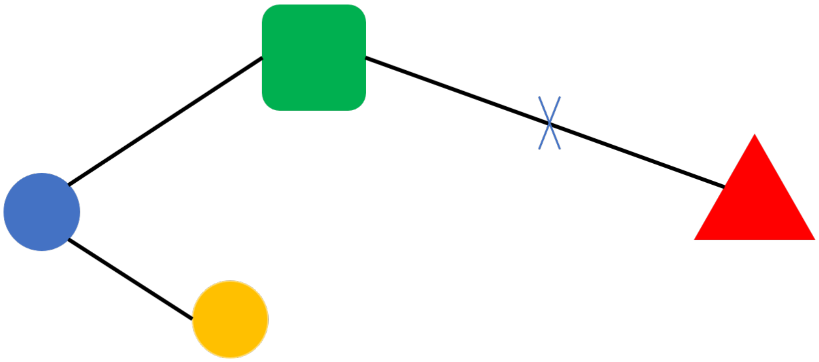


Figure 1.2: There are four entities in this group, each entity has its appearance and geometric fluent. These four entities together with their connections define the topology of the group. Cutting the connection between triangle and square change the topology of the group, making it into two groups.

Disentangling the fluents change can be beneficial because different objects share the same fluents change space like brightness, rigid body motion, splitting although the objects are coming in various shapes, sizes and colors. Under certain human actions such as opening, pulling or physics law like conservation of momentum, different objects will change following the same pattern. With this we can:

- Recognize tasks from the object fluents change which are involved
- Predicting the future of objects after certain actions to help planning the task

Directly disentangling the fluents change from the raw images are difficult since they all contain the change of pixels. We seek to use a generative network to encode the fluents of objects into a latent space which can be disentangled into appearance, geometry and topology.

We will use a generative model to learn the fluent space. The generative model is of the following form $X = g(x; W) + \epsilon$, where X is the observed image of dimension D, x is the latent vector of dimension d in fluent space. x has a prior distribution $N(0, I_d)$, and $\epsilon \sim N(0, \sigma^2 I_D)$ is the noise, where I_d, I_D denote the identity matrix.

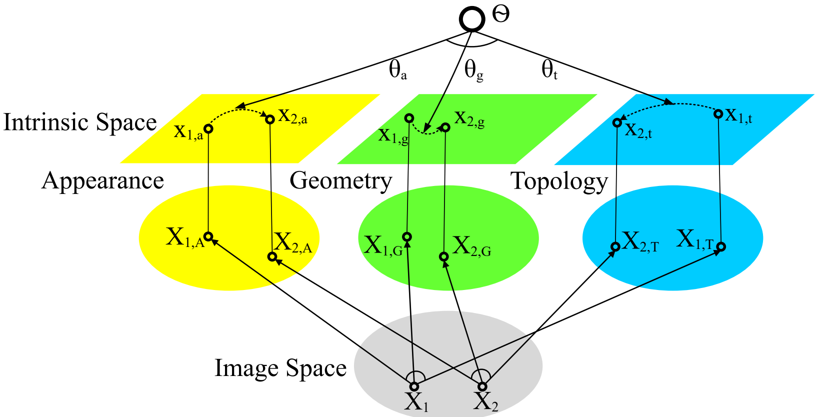


Figure 1.3: Illustration of fluent changes of two images in fluent space. Image X_1 and X_2 are firstly disentangled as appearance, geometry and topology. Fluent changes are represented as the parameters that control the changing process in each latent space.

We can use the generator network to generalize the model. Firstly, we use the following network to model one category image:

$$X_A = g_a(x_a; W_a); X_G = g_g(x_g; W_g); X_T = g_t(x_t; W_t); \tag{1.1}$$

$$X = \omega(X_A; X_G; X_T) + \epsilon. \tag{1.2}$$

We assume that the appearance vector x_a, the geometry vector x_g and the topology vector x_t are to be independent. This disentangles appearance, geometry and topology variations. The appearance vector x_a generates the canonical appearance (image) X_A. The geometry vector x_g generates the shape (deformation) X_G. The topology vector x_t generates the topology (structure) X_T. The final image X is obtained by applying the shape deformation X_G and topological change X_T to the canonical image X_A, plus additive noise ϵ.

For fluents recognition in fluent space, suppose the images before/after an action are X_1 and X_2. Their latent representations in appearance, geometry and topology space are $x_{1,a}, x_{1,g}, x_{1,t}$ and $x_{2,a}, x_{2,g}, x_{2,t}$ respectively. Fluent changes in each intrinsic space are defined as below:

$$x_{2,a} = f_a(x_{1,a}; \theta_a); x_{2,g} = f_g(x_{1,g}; \theta_g); x_{2,t} = f_t(x_{1,t}; \theta_t). \tag{1.3}$$

Our goal is to recover the fluent changes in these three intrinsic spaces as shown in Figure 1.3. Fluent changes are represented by parameters $\Theta = (\theta_a, \theta_g, \theta_t)$ that control the fluents changing process such as brightness, translation distance and rotation angle.

1.2 Mutual Fluents and Task Representation

In the last section, we have defined self fluents of objects which are the properties of entity which can be an individual or a group. Mutual fluents describe the variable states of relationship between two entities, including visible fluents such as the relative position between hand and tool and invisible fluents like human's attention on task-related objects.

To do a task is to achieve certain goal, which can be represented by a set of fluents change. For example, the fluents change of making a coffee can be represented by a cup full of coffee which is empty before the task. However this kind of fluent change is impossible to be achieved by one step with just one empty mug. We need a physically feasible way to turn a empty cup to a cup of coffee, like first pouring the coffee powder into the cup, and then fetch the water from dispenser. In another word, we need a plan to decompose making coffee into several steps which represents making coffee as an STC-AOG, as illustrated in Figure 1.4

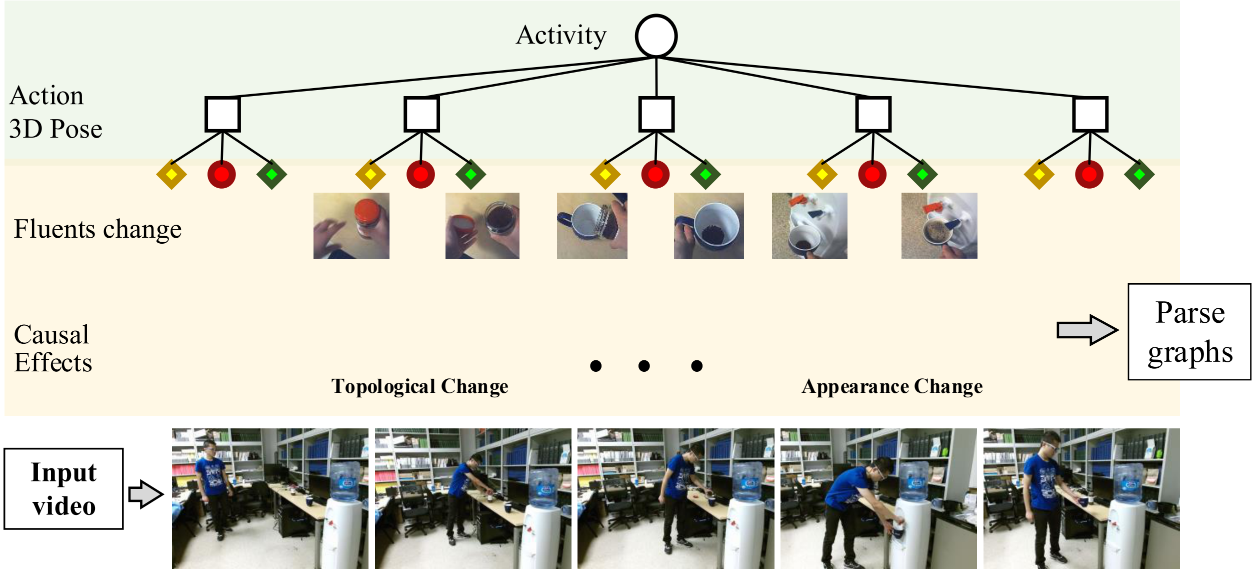


Figure 1.4: Illustration of the parse graph to make a coffee, which can be represented by a series of fluents change.

In above figure, the 3D pose describe the spatial mutual fluents between human and the objects. After each step, the self fluents of the involving objects will change like the topological change for the coffee can. We use And-Or-Graph(AOG) to represent the task is because we have many different ways to achieve the final fluents change, while each feasible series of steps is a parse graph(PG).

CHAPTER 2

Learning Objects and Attributes Relation in Latent Space

2.1 Introduction

Attributes are the description of ‘objects’. To reach a higher level of vision understanding, we hope that a computer can understand not only object categories but also their attributes. As a result, recognizing objects with their attributes have been widely studied in various problems such as person re-identification [SZX16], scene understanding [LRT14], image caption [WSW17], image search [KBN08], and image generation [YYS16].

Encouraged by the great success of the discriminative model implemented with deep neural networks for object classification [SZ14b, HZR16, HLV17], some studies like [MGH17] have tried to recognize attribute-object pairs by composing the discriminative models that are separately trained for the object and attribute. Factually, the discriminative models are trying to learn the attribute visual ‘prototype’ and object visual ‘prototype’. It is true that an object always has a visual ‘prototype’. For example, when we ask people to draw a dog, different people may draw beagles, collies, dalmatians or poodles, but the ‘dogs’ they draw always have two ears and four legs. However, if we ask people to draw ‘old’, people may find difficulties to do because the ‘old’ is nonobjective and does not present clear visual ‘prototype’. As illustrated in the upper row of Figure 2.1, the dogs with different attributes present the similar visual feature. On the contrary, as shown in the lower row, the visual feature of attribute ‘old’ varies dramatically for different object classes. As a result, the discriminative model is not such successful for attribute recognition as that for object recognition, which further results in the low accuracy of attribute-object pair recognition.

In fact, attribute is highly dependent on object. For example, when we teach a baby to recognize the attribute 'old', we often use instances like 'old book', 'old bike', and 'old dog' to show how 'old' looks like. Therefore, to better recognize attribute-object pair, we should explore the inner relation of the attribute and object instead of composing the discriminative models that are separately trained for the object and attribute.

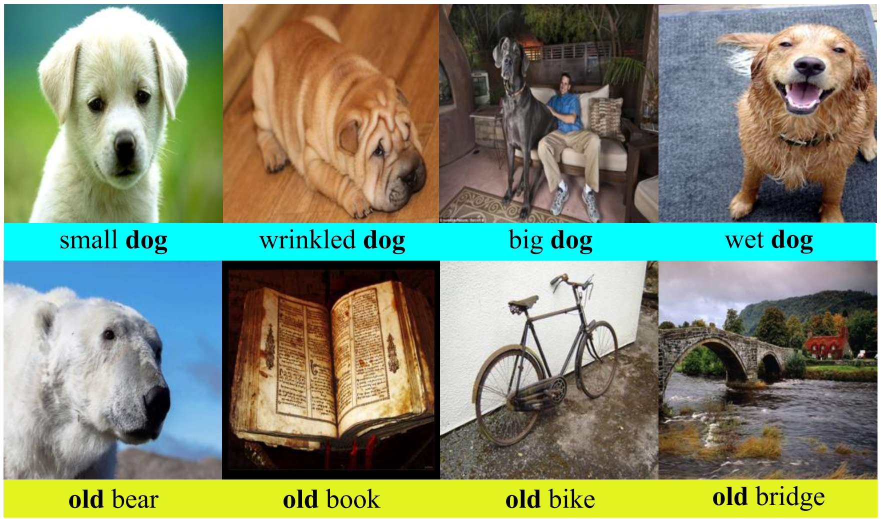


Figure 2.1: The upper row shows the same object 'dog' with the different attributes. Though dogs can be small, big, wet, or wrinkled, they present similar visual properties. The lower row shows the same attribute 'old' with the different objects, we can observe that the visual properties greatly differ from each other.

Realizing this issue, some works like [CG14] have tried to process the attribute and object as a whole to explore their inner relations. However, many of them still employ discriminative models to tackle the problem, resulting in poor performance for recognizing unseen attribute-object pairs. The major reason is that the individual property of the attribute and object are not learned. For example, there are pairs like 'old book' and 'small dog' in the training set, so the model are well fitted to these pairs. However, the model fails to learn the concept of individual attribute and object like 'old', 'book', 'small', and 'dog', thus can hardly generalize to unseen pairs like 'old dog' or 'small book'.

Summing up the above, to recognize unseen attribute-object pairs, we should design a model that should consider not only the individual property of the attribute and object but also the inner relation between them. To this end, in this work, we propose an encoder-decoder generative model bridging visual and linguistic features in a unified end-to-end network. We first obtain the visual feature of images using state-of-the-art deep neural network, and the linguistic feature by extracting the semantic word embedding vectors of object and attribute label. To explore the inner relation of the attribute and object, inspired by the idea of ZSL (Zero-Shot Learning), we project the visual feature and the linguistic feature into a latent space where the attribute and object are processed as a whole. During the projection, to preserve the individual property of the attribute and object, the original visual and linguistic features of the attribute and object are projected by different functions. In the latent space, in order to minimize the 'distance' between the visual feature and linguistic feature, we have exploited several loss functions to penalize the dissimilarity between them. In addition, we propose the decoding loss which has been proved crucial to generalize better to unseen pairs because it can find the natural and intrinsic feature representations.

In the experiments, we have compared with four state-of-the-art methods on two challenging datasets, MIT-States [ILA15] and UT-Zappos50K [YG14]. The experiments show that our method outperforms other methods significantly. We also performed some ablation experiments to study the effect of individual loss function, the influence of visual feature extractor, and the interdependence of the attribute and object, from which we draw some important conclusions.

our contributions in this work are as follows:

- We propose a generative model to recognize unseen attribute-object pairs instead of composing multiple classifiers.

- Our model combines the visual and linguistic information in the same latent space, which is significant for exploring the inner relation of the attribute and object.

- We apply the encoder-decoder mechanism to the problem of attribute-object pair recognition.

2.2 Related Work

Object and attribute. In recent years, object, attribute, and attribute-object composition recognition have been intensively studied in both image and video domains [LRT14, ILA15, SZX16, WSW17, MGH17, LWZ17]. Attribute recognition is a basic problem, the typical method for attribute recognition is similar to that for object classification, training discriminative models using attribute-labeled samples [PG11, PH12, SKB12, SL16, LKZ17]. Attribute-object pair recognition is a more challenging problem. Some conventional methods often use a classifier or compose multiple classifiers to tackle the problem [CG14, MGH17]. Some other studies assume object-attribute relationship is known and datasets are simple that only contain one or few dominant object categories [WM10, MSN11, WJ13]. To make the model applicable to complex datasets that cover various object and attribute classes, some good models are proposed [WJW13, KPO13]. However, they are suffering from 'domain shift' problem - the test data distribution differs from that estimated by the training data, leading to the low performance on testing data. To overcome this problem, the work [NG18] proposes to take the attribute as the operator and attribute-object pair as a vector that is transformed by this operator, then this transformed vector is compared with CNN visual feature to recognize unseen pairs. In this work, we propose a generative model with encoder-decoder mechanism which is significant for exploring the intrinsic feature representation, thus can better transfer the concept of object and attribute from training set to the testing set.

Zero-shot learning. The goal of zero-shot learning (ZSL) is to recognize unseen/new objects by utilizing their auxiliary information such as attribute or text description. One major method for zero-shot learning is first mapping the input into a semantic space where the auxiliary information like attributes of unseen objects are known, then finding the object whose auxiliary feature is 'closest' to the input feature [LNH14, ARW15]. Another major method learns a latent space that the input and the auxiliary feature of unseen objects are simultaneously projected into [CCG16, WCC16], and the most likely unseen object is recognized by measuring the 'distance' between input feature and auxiliary feature in the latent space. Some other methods predict unseen objects using the classifier that is composed by seen object classifiers [NMB13, CCG16]. Recently, semantic autoencoder (SAE) has been proposed for zero-shot learning, considering both projection from input space to semantic space and the reverse, which demonstrates to be a simple but effective approach [KXG17]. Several other works like [WPV18] have explored more general generative methods using highly nonlinear model instead of linear regression from the latent space to the input space [VAM18]. Inspired by these works, in this work, we project both visual and linguistic features into the same latent space where the most likely attribute-object pair is selected with the least loss calculated by our self-defined loss function.

Vision and language combination. With the rapid development of vision and language, vision and language combination has been studied to tackle many problems. For example, as mentioned above, many ZSL models take linguistic text description [LSF15, ESE13] as auxiliary information for unseen object recognition. However, text description annotation is 'expensive', especially for large scale datasets. Therefore, it is intuitive to utilize linguistic word embedding as auxiliary information because all words can be encoded as vectors with the pre-trained model [SGM13, FCS13]. In this work, we represent object and attribute as linguistic word embedding vectors. Different from one-hot vectors, word embedding vectors imply the semantic similarity of their corresponding words. In another word, semantically similar attribute and object will create similar word embedding vectors, which is significant for learning the inner relation of the attribute and object.

2.3 Approach

In this work, we are studying the problem of identifying the attribute-object pair of the given image. For example, given an image as shown in Figure 2.2, our task is to output the attribute-object pair 'wrinkled dog'. It is challenging for two reasons: 1) we are recognizing unseen attribute-object pairs that are not included in training data, and 2) this is fine-grained recognition problem and the number of possible attribute-object pairs is large.

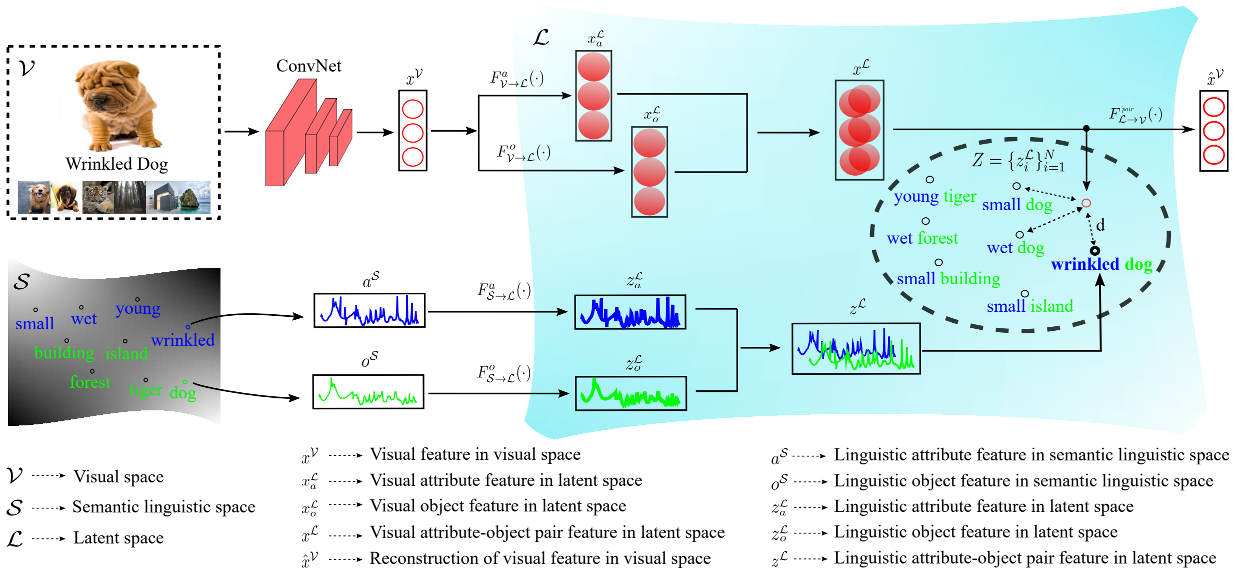

$\mathcal{V}$
Wrinkled Dog
ConvNet
$x^{\mathcal{V}}$
$\mathcal{L}$
$x_a^{\mathcal{L}}$
$F^a_{\mathcal{V}\to\mathcal{L}}(\cdot)$
$F^o_{\mathcal{V}\to\mathcal{L}}(\cdot)$
$x_o^{\mathcal{L}}$
$x^{\mathcal{L}}$
$F^{pair}_{\mathcal{L}\to\mathcal{V}}(\cdot)$
$\hat{x}^{\mathcal{V}}$
$Z = \{z_i^{\mathcal{L}}\}_{i=1}^N$
young tiger
small dog
wet forest
wet dog
d
wrinkled dog
small building
small island
$\mathcal{S}$
small
wet
young
wrinkled
building
island
forest
tiger
dog
$a^{\mathcal{S}}$
$o^{\mathcal{S}}$
$F^a_{\mathcal{S}\to\mathcal{L}}(\cdot)$
$F^o_{\mathcal{S}\to\mathcal{L}}(\cdot)$
$z_a^{\mathcal{L}}$
$z_o^{\mathcal{L}}$
$z^{\mathcal{L}}$
$\mathcal{V}$ → Visual space
$\mathcal{S}$ → Semantic linguistic space
$\mathcal{L}$ → Latent space
$x^{\mathcal{V}}$ → Visual feature in visual space
$x_a^{\mathcal{L}}$ → Visual attribute feature in latent space
$x_o^{\mathcal{L}}$ → Visual object feature in latent space
$x^{\mathcal{L}}$ → Visual attribute-object pair feature in latent space
$\hat{x}^{\mathcal{V}}$ → Reconstruction of visual feature in visual space
$a^{\mathcal{S}}$ → Linguistic attribute feature in semantic linguistic space
$o^{\mathcal{S}}$ → Linguistic object feature in semantic linguistic space
$z_a^{\mathcal{L}}$ → Linguistic attribute feature in latent space
$z_o^{\mathcal{L}}$ → Linguistic object feature in latent space
$z^{\mathcal{L}}$ → Linguistic attribute-object pair feature in latent space

Figure 2.2: Given an image with the attribute-object pair label ‘wrinkled dog’, our goal is to correctly recognize this label. The challenge is that there are many possible pairs. As shown in the black ellipse in latent space $\mathcal{L}$, the pair may be ‘young tiger’, ‘small dog’, or others. To find the correct one, we first extract the visual feature $x^{\mathcal{V}}$ of the given image in visual space $\mathcal{V}$ and project $x^{\mathcal{V}}$ into latent space $\mathcal{L}$. After some processing, we finally obtain $x^{\mathcal{L}}$ that represents the visual feature of the given image in $\mathcal{L}$. At the same time, for all possible pairs, we extract their linguistic representations $Z = \{z_i^{\mathcal{L}}\}_{i=1}^N$ in latent space $\mathcal{L}$. Each element $z_i^{\mathcal{L}}$ in Z corresponds to one possible pair, which is obtained by mapping the corresponding word embedding vectors $a^{\mathcal{S}}$ and $o^{\mathcal{S}}$ from semantic linguistic space $\mathcal{S}$ to latent space $\mathcal{L}$. Finally, we take the label of $z_i^{\mathcal{L}}$ with closest distance to $x^{\mathcal{L}}$ as recognition result.

2.3.1 Overview

Given an image I with the attribute label y_a and object label y_o, our goal is to correctly choose its attribute-object label from the set $Z = \{z_i^{\mathcal{L}}\}_{i=1}^N$ that contains all possible N attribute-object pairs. To realize this goal, the intuitive idea is combining the classifiers that are separately trained for the attribute and object. For example, to recognize unseen attribute-object pair ‘small dog’, some studies first learn the concept of ‘small’ from images like ‘small cat’, ‘small horse’, and other small objects as well as the concept of ‘dog’ from images like ‘wrinkled dog’, ‘big dog’ and other dogs using training set, and then combine the separate classifiers to recognize unseen attribute-object pair 'small dog'. However, as we have discussed previously that the attribute does not have clear visual 'prototype' and is highly dependent on the object. Therefore, we propose a generative model that combine the visual and linguistic information in the same latent space where the attribute and object are processed as a whole. As shown in Figure 2.2, We first use deep neural networks [SZ14b, HZR16] to extract the visual feature of I in visual space $\mathcal{V}$ and denote it as $x^{\mathcal{V}}$, which is then projected into latent space $\mathcal{L}$ as $x^{\mathcal{L}}$. For all possible pairs, we extract their attribute vector $a^{\mathcal{S}}$ and object vector $o^{\mathcal{S}}$, and then project them into latent space $\mathcal{L}$ where they are merged as linguistic attribute-object pair feature $z^{\mathcal{L}}$. In the training stage, we are trying to learn the projection from $\mathcal{V}$ and $\mathcal{S}$ to $\mathcal{L}$ by minimizing the 'distance' between the visual feature $x^{\mathcal{L}}$ and its corresponding linguistic feature $z^{\mathcal{L}}$ in the latent space. In the testing stage, we predict the attribute-object label of $z^{\mathcal{L}}$ that is the closest to the $x^{\mathcal{L}}$. To this end, we need to consider two crucial problems: 1) how is original visual and linguistic data transitioned between different spaces to obtain $x^{\mathcal{L}}$ and $z^{\mathcal{L}}$, and 2) how to design the loss functions to minimize their 'distance'. We will tackle these two problems in the following sub-sections.

2.3.2 Data transitions

Figure 2.2 signals the data transition process of our method. In the figure, the red circles represent the visual data flow, while green and blue waves represent the linguistic data flow. For visual data flow, the visual feature $\mathbf{x}^{\mathcal{V}}$ in visual space $\mathcal{V}$ is projected into latent space $\mathcal{L}$ as two flows representing visual attribute feature $x_a^{\mathcal{L}}$ and visual object feature $x_o^{\mathcal{L}}$ respectively. $x_a^{\mathcal{L}}$ and $x_o^{\mathcal{L}}$ are then merged as visual attribute-object pair feature $x^{\mathcal{L}}$. To obtain reconstruction of the original visual feature, $x^{\mathcal{L}}$ is re-mapped back to visual space $\mathcal{V}$, the re-mapped feature is denoted as $\hat{x}^{\mathcal{V}}$. For linguistic data flow, the linguistic attribute feature $a^{\mathcal{S}}$ and linguistic object feature $o^{\mathcal{S}}$ in word embedding space $\mathcal{S}$ are projected into the latent space as $z_a^{\mathcal{L}}$ and $z_o^{\mathcal{L}}$, respectively. Then $z_a^{\mathcal{L}}$ and $z_o^{\mathcal{L}}$ are merged as linguistic attribute-object pair feature $z^{\mathcal{L}}$.

For the projections of linguistic attribute and object features from $\mathcal{S}$ to $\mathcal{L}$, we define two projection functions $F^a_{\mathcal{S}\to\mathcal{L}}(\cdot)$ and $F^o_{\mathcal{S}\to\mathcal{L}}(\cdot)$ as linguistic encoders:

$$z^{\mathcal{L}}_a = F^a_{\mathcal{S}\to\mathcal{L}}(a^{\mathcal{S}}) \tag{2.1}$$

$$z^{\mathcal{L}}_o = F^o_{\mathcal{S}\to\mathcal{L}}(o^{\mathcal{S}}) \tag{2.2}$$

Here, we define different projection functions for the attribute and object because object (a noun in most cases) and attribute (an adjective to describe the noun in most cases) are two different kinds of instances and should be processed differently to explore their potential properties. The experiment results validate the effectiveness of this separate processing, the details can be found in the experiment section.

Based on the above two transitions, we can get the attribute feature and object feature individually. To explore their inner relation, we add them together:

$$z^{\mathcal{L}} = z^{\mathcal{L}}_a + z^{\mathcal{L}}_o \tag{2.3}$$

For visual feature in $\mathcal{V}$, we define two project functions $F^a_{\mathcal{V}\to\mathcal{L}}(\cdot)$ and $F^o_{\mathcal{V}\to\mathcal{L}}(\cdot)$ which project $x^{\mathcal{V}}$ to its attribute feature $x^{\mathcal{L}}_a$ and object feature $x^{\mathcal{L}}_o$.

$$x^{\mathcal{L}}_a = F^a_{\mathcal{V}\to\mathcal{L}}(x^{\mathcal{V}}) \tag{2.4}$$

$$x^{\mathcal{L}}_o = F^o_{\mathcal{V}\to\mathcal{L}}(x^{\mathcal{V}}) \tag{2.5}$$

With the same manner as we combine latent attribute feature and object feature from language. We get $x^{\mathcal{L}}$ by:

$$x^{\mathcal{L}} = x^{\mathcal{L}}_a + x^{\mathcal{L}}_o \tag{2.6}$$

Our decoder to re-map the visual feature from latent space $\mathcal{L}$ to original visual space $\mathcal{V}$ is defined as:

$$\hat{x}^{\mathcal{V}} = F^{pair}_{\mathcal{L}\to\mathcal{V}}(x^{\mathcal{L}}) \tag{2.7}$$

2.3.3 Loss functions

2.3.3.1 Encoding loss

The goal of encoding loss is to minimize the 'distance' between visual attribute-object pair feature $x^{\mathcal{L}}$ and linguistic attribute-object pair feature $z^{\mathcal{L}}$. There are multiple ways to define the distance in the latent space. The most intuitive and common one is L2 distance. However, when using L2 distance, the value of visual and linguistic features in latent space tend to be extremely small during optimization which leads to poor performance. Therefore, we define the encoding loss as cosine similarity between $x^{\mathcal{L}}$ and $z^{\mathcal{L}}$:

$$L_{en} = dist(x^{\mathcal{L}}, z^{\mathcal{L}}) = 1 - \frac{< x^{\mathcal{L}}, z^{\mathcal{L}} >}{\|x^{\mathcal{L}}\|_2 \|z^{\mathcal{L}}\|_2}, \tag{2.8}$$

From geometrical aspect, we measure the angle between two vectors, and encourage them to have the same direction. The reason we do not consider the cosine similarity between $x_o^{\mathcal{L}}$ and $z_o^{\mathcal{L}}$ as well as the similarity between $x_a^{\mathcal{L}}$ and $z_a^{\mathcal{L}}$ is that individual attribute or object feature includes limited information for attribute-object pair estimation. In another word, we treat the attribute and object in the latent space as a whole to explore their inner relation.

2.3.3.2 Triplet loss

The encoding loss defined in the Eq. 2.8 encourages the visual feature to be close to the linguistic feature of the indexed attribute-object pair, but doesn't consider that the visual feature should be far from the linguistic feature of other attribute-object pairs. Therefore, we add an extra loss called triplet loss, which impels the distance between $x^{\mathcal{L}}$ and $z^{\mathcal{L}}$ to be smaller than the distance between $x^{\mathcal{L}}$ and other linguistic attribute-object features $\tilde{z}^{\mathcal{L}}$ by a margin K:

$$L_{tri} = max\left(0, \frac{dist(x^{\mathcal{L}}, z^{\mathcal{L}})}{dist(x^{\mathcal{L}}, \tilde{z}^{\mathcal{L}})} - K\right) \tag{2.9}$$

2.3.3.3 Decoding loss

Inspired by recent zero-shot learning works using autoencoder [KXG17, WPV18], to explore the intrinsic representation of the input image, we introduce the decoding loss which is defined as the L2 distance between original visual feature $x^{\mathcal{V}}$ and reconstructed visual feature $\hat{x}^{\mathcal{V}}$:

$$L_{de} = \|x^{\mathcal{V}} - \hat{x}^{\mathcal{V}}\|_2 \tag{2.10}$$

The decoding loss encourages $\hat{x}^{\mathcal{V}}$ to be same with $x^{\mathcal{V}}$ rather than minimizing the angle between $\hat{x}^{\mathcal{V}}$ and $x^{\mathcal{V}}$. Therefore, we utilize the L2 loss instead of cosine similarity. We did not apply the decoding loss to linguistic features. The reason is that one attribute-object class only corresponds to one linguistic vector, the number of attribute-object classes is too small to learn a re-projection function with a huge number of parameters.

2.3.3.4 Discriminative loss

In the above, we have introduced the encoding loss to encourage the visual attribute-object pair feature to be close to the indexed linguistic attribute-object pair feature. However, this may lead to that the dominance effect of attribute or object. In another word, either attribute or object tends to represent the whole pair. To avoid this imbalance case, we define the discriminative loss to encourage to preserve the information for attributes and objects. The discriminative loss consists of attribute discriminative loss and object discriminative loss:

$$L_{dis} = L_{dis,a} + L_{dis,o} \tag{2.11}$$

$L_{dis,a}$ and $L_{dis,o}$ are defined as

$$L_{dis,a} = h(x_a^{\mathcal{L}}, y_a), \quad L_{dis,o} = h(x_o^{\mathcal{L}}, y_o)$$

where $h(\cdot)$ is a one fully connected layer network with cross-entropy loss.

The purpose of discriminative loss is to stress the individual property of the attribute and object in visual domain. The experiments validate the effectiveness of the discriminative loss, the details can be found the in experiment section.

2.3.4 Learning and inference

The purpose of learning is to estimate the parameters of data transition functions defined in Eq. 2.1-2.7. Let W be all parameters of functions involved in Eq. 2.1-2.7. Given a set of images with the attribute and object labels, the estimation of W is equal to minimize the losses defined in Eq. 2.8-2.11 where κ,α, β, γ are the weights for different losses:

$$W^* = \arg\min_W(\kappa L_{en} + \alpha L_{tri} + \beta L_{de} + \gamma L_{dis}) \tag{2.12}$$

During inference, when a new image arrives, the visual feature is extracted and then projected to latent space, producing the visual representation $x^{\mathcal{L}}$ in latent space. At the same time, the linguistic features of all N possible attribute-object pairs are also computed to obtain a set of linguistic representations $Z = \{z_i^{\mathcal{L}}\}_{i=1}^{N}$. We compute the cosine similarity between $x^{\mathcal{L}}$ and every $z_i^{\mathcal{L}}$, and select the indexed attribute-object pair label of the most similar $z_i^{\mathcal{L}}$ as recognition result. In another word, we predict the attribute-object pair label by choosing the $z_i^{\mathcal{L}}$ with the least encoding loss.

Methods	MIT-States(%)	UT-Zappos(%)
CHANCE	0.14	3.0
ANALOG[CG14]	1.4	18.3
SAE [KXG17]	14.0	31.0
REDWINE [MGH17]	12.5	40.3
ATTOPERATOR [NG18]	14.2	46.2
Ours	**17.8**	**48.3**

Table 2.1: Top-1 accuracy of methods tested on the MIT-States dataset and UT-Zappos50k dataset. For fair comparison, all methods use the same visual feature extracted with ResNet-18.

2.4 Experiments

2.4.1 Setup

2.4.1.1 Datasets

Two datasets, MIT-States [ILA15] and UT-Zappos50K [YG14], are used for evaluation. The MIT-States is a big dataset with 63,440 images. Each image is annotated with an attribute-object pair such as 'small bus'. It covers 245 object classes and 115 attribute classes. However, it does not have 245×115 attribute-object pairs as labels because not all pairs are meaningful in real world. Following the same setting as in [MGH17] and [NG18], 1,262 attribute-object pairs are used for training and 700 pairs for test. The training pairs and testing pairs are non-overlapping. UT-Zappos50k is a fine-grained shoe dataset with 50,025 images. Following the same setting as in [NG18] We use 83 attribute-object pairs for training and 33 pairs for testing. The training pairs and testing pairs are also non-overlapping.

2.4.1.2 Baselines and metric

We widely compare with four baseline methods, three of them are recently proposed state-of-the-art methods. ANALOG [CG14] predicts unseen attribute-object pairs using a sparse set of seen object-specific attribute classifiers. SAE [KXG17] predicts unseen pairs by projecting the input feature in a semantic space where the auxiliary information of unseen pairs is known. REDWINE [MGH17] predicts unseen attribute-object pairs by composing existing attribute and object classifiers. ATTOPERATOR [NG18] represents the attribute-object pair as the object vector transformed by attribute operator, the transformed vector is compared with CNN visual feature for unseen pair recognition. We use the top-1 accuracy on testing images as evaluation metric.

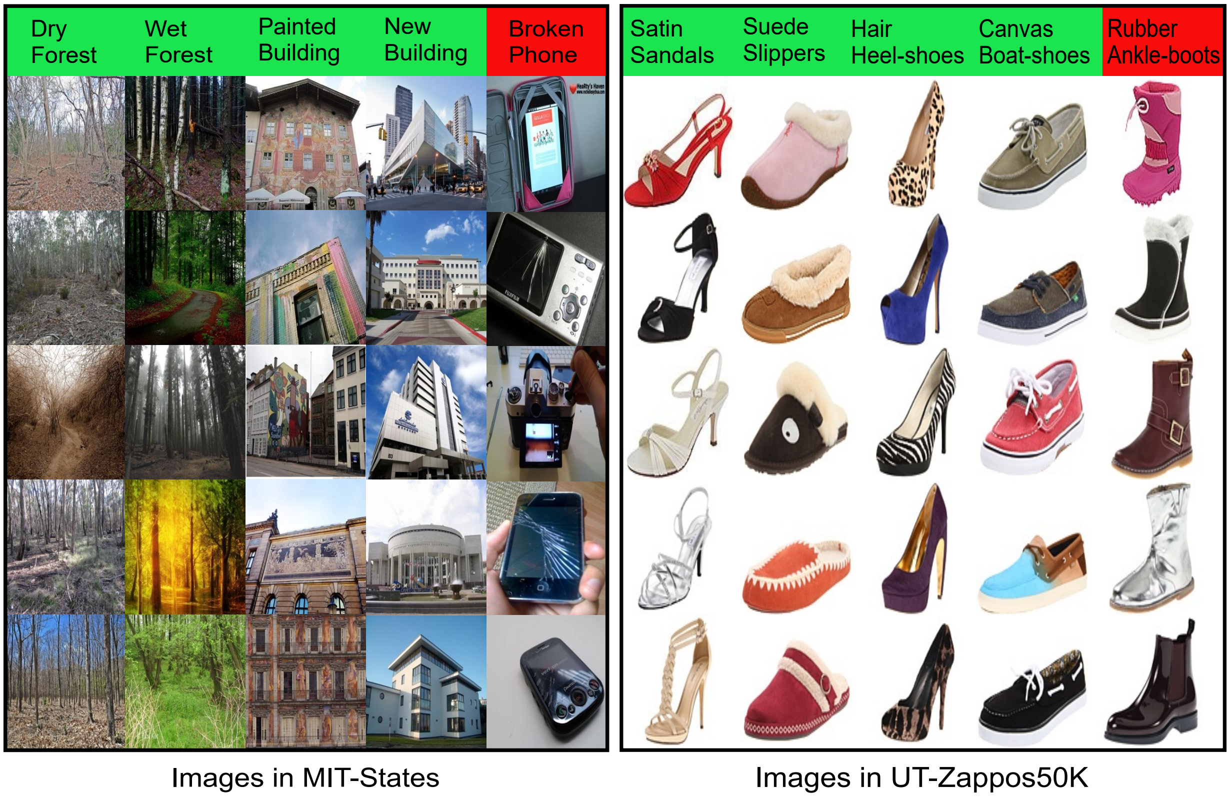
2.4.1.3 Implementation details

We extract 512 dimension visual feature of the image using ResNet-18 [HZR16] pre-trained on ImageNet [RDS15]. The network is not fine-tuned on MIT-States or UT-Zappos50K dataset. We extract 300 dimension linguistic feature for object and attribute using pre-trained word vectors [PSM14], some not-included words are substituted by synonyms. All these features are mapped into a 1024 dimension latent space. K in Eq. 2.9 is a parameter that controls the margin, and is set to 0.9 in our experiment. κ,α, β, γ in Eq. 6.8 are with the ratio of $1 : 0.2 : 2 : 2$ for Mit-States dataset and $1 : 0.2 : 0.5 : 2$ for UT-Zappos50K dataset. We implement our end to end neural network with MXNet [CLL15]. For all the projection functions, we implement each as one fully connected layer. For the projections from visual space to latent space, to resolve overfitting problem, we add dropout layers after each fully connected layer with dropout ratio as 0.3. We use ADAM as our optimizer with the initial learning rate as 0.0001, which decays by 0.9 every two epochs. At every iteration we feed the mini-batch to the network with the batch size as 128.

2.4.2 Quantitative results

As shown in Tab 2.1, our method outperform all recently proposed methods, achieving 25.4% and 4.5% improvement over the second best methods respectively on MIT-States and UT-Zappos50k datasets. Our method outperforms others for two reasons: 1) we introduce the encoder-decoder mechanism that enables to learn the general and intrinsic representation of attributes-object pair, and 2) our model considers not only the individual property of the attribute and object but also the inner relation between them.

The average accuracy on UT-Zappos50k is higher than that on MIT-States. This mainly results from the difference of data complexity. Images in UT-Zappos50k have single white background as shown in Figure 2.3, and few attribute and object classes, while images in MIT-States cover a variety of backgrounds, object classes, and attribute classes. In addition, only 33 attribute-object pairs are used for testing in UT-Zappos50k, while 700 pairs are used in MIT-States.



Images in MIT-States

Images in UT-Zappos50K

Figure 2.3: Qualitative results on two datasets. For each dataset, the fifth column shows some false recognitions and other columns show the true recognitions.

2.4.3 Qualitative results

Figure 2.3 shows some qualitative results on MIT-States dataset and UT-Zappos50K dataset. Columns with the green mark show some true recognitions. We can observe that some samples with extremely abstract attribute-object pairs are correctly recognized. For example, some images like ‘wet forest’ and ‘dry forest’ are correctly recognized. However, the accuracy is low. On one hand, forest is an abstract object sharing similar properties with objects like tree, bush, plant, and jungle. On the other hand, wet and dry are abstract attributes sharing the similar properties with attributes like damp, verdant, mossy, and barren. Factually for some ‘wet forest’ testing images, our model has recognized them as ‘damp bush’, ‘mossy jungle’, ‘verdant plant’ and other similar compositions. This demonstrates that our model has learned some macro concepts for attribute-object pairs, but some micro concepts have not been precisely distinguished.

Columns with the red mark shows some false recognitions. We can observe that some images with the label of ‘scratched phone’ and ‘broken camera’ are wrongly recognized as ‘broken phone’, and pairs like ‘synthetic ankle-boot’ are wrongly recognized as ‘rubber ankle-boot’. One main reason is that some attributes are similar and some objects present similar appearance. For example, the attribute of ‘scratched’ is similar with ‘broken’, the appearance of some cameras is similar to that of phone, and some ‘synthetic’ boots also present the attribute of ‘rubber’.

2.4.4 Attribute and object relation

To validate the relation of the attribute and object we have learned, we designed an experiment that measures the attribute and object recognition accuracy under two conditions: 1) not considering the relation of the attribute and object, as ‘Att’ and ‘Obj’ shown in Table. 2.2, and 2) considering the relation of the attribute and object in both visual and linguistic domains, as ‘Att (pair)’ and ‘Obj (pair)’ shown in Table. 2.2. Actually, ‘Att (pair)’ and ‘Obj (pair)’ correspond to the accuracy that are extracted from our attribute-object pair recognition results. While for ‘Att’ and ‘Obj’, we have separately trained a 2-layer MLP model to recognize attribute and object category. We can observe from Table. 2.2 that attribute recognition accuracy of our model is always higher than that does not consider relation of the attribute and object, which validates our claim that attribute is highly dependent on object.

2.4.5 Ablation study

We design two experiments. One is to study the importance of different loss functions, the other is to study the effect of different visual features.

For the detail analysis of different loss functions, we report the accuracy corresponding to different kinds of loss function compositions on both MIT-States and UT-Zappos50K datasets as shown in Table. 2.3. If only encoding loss is used, the accuracy is 3.6% on the MIT-States and 37.8% on the UT-Zappos50K. If we add triplet loss (+tri) to encoding loss,

Item	MIT-States		UT-Zappos	
	Top1 (%)	Top5(%)	Top1 (%)	Top5(%)
Att	15.1	38.9	18.4	76.8
Att (pair)	**25.1**	**55.3**	**52.0**	**92.7**
Obj	27.7	**56.4**	68.1	**96.7**
Obj (pair)	**29.9**	51.6	**77.3**	93.9

Table 2.2: 'Att' and 'Obj' correspond to the accuracies that are trained using the original visual feature, while 'Att (pair)' and 'Obj (pair)' correspond to the accuracies that are extracted from our attribute-object pair recognition results.

we obtain significant performance improvement on both datasets, which validate our claim that we should not only encourage the visual feature to be close to the linguistic feature of its indexed attribute-object pair but also should let it to be away from the linguistic features of the other attribute-object pairs. If we add discriminative loss (+dis) to encoding loss, we obtain significant performance improvement on MIT-States but slight improvement on UT-Zappos50K. On one hand, we can conclude that the discriminative loss allows to learn better visual attribute-object representation by stressing individual attribute and object property. On the other hand, it is based on certain condition. The MIT-States dataset is complex and has many object and attribute classes, while the UT-Zappos50K is relatively simple that the visual features already have good representations, so adding discriminative loss only contributes slightly. If we add decoding loss (+de), we obtain impressive performance improvement on the MIT-States, even better than adding both triplet loss and discriminative loss (+tri+dis), which demonstrates that encoder-decoder mechanism can mine the essential representation for attribute-object pair. On the UT-Zappos50K, the decoding loss is also helpful.

From the Table. 2.3 we can observe that the performance is basically increasing when adding more loss functions. Though in some cases adding more losses lead to worse results, the best performance is achieved when using all loss functions (+de+dis+tri), from which we can conclude that the four loss functions are complementary to each other. In the Table.

Loss	MIT-States(%)	UT-Zappos(%)
en	3.6	37.8
+tri	11.2	45.5
+dis	15.3	37.9
+de	17.2	41.3
+tri+dis	15.7	41.4
+tri+de	17.4	46.7
+dis+de	17.5	43.5
+de+dis+tri*	16.5	47.5
+de+dis+tri	**17.8**	**48.3**

Table 2.3: Accuracy for loss function ablation study. 'en', 'tri', 'dis', and 'de' represent the encoding loss, triplet loss, discriminative loss, and decoding loss, respectively. *means sharing parameters for $F^{a}_{\mathcal{S}\to\mathcal{L}}(\cdot)$ and $F^{o}_{\mathcal{S}\to\mathcal{L}}(\cdot)$

2.3, (+de+dis+tri*) corresponds to the accuracy when we impose the constraint that the attribute and object are processed by the same projection function (sharing the parameters in Eq. 2.1 and Eq. 2.2). We can observe that the accuracy of (+de+dis+tri*) is lower than that of (+de+dis+tri) on both datasets, from which we can draw another important conclusion that object and attribute are two different kinds of instances and should be processed differently to better explore their individual properties. .

In Table. 2.4, we report the accuracy corresponding to different kinds of visual feature extractors. We tested two kinds of network architectures, VGG [SZ14b] and ResNet [HZR16]. We can observe from table that the visual feature significantly affects the final performance. VGG-19 presents similar performance with VGG-16. ResNet behaves better than VGG, and basically achieves higher accuracy with deeper architectures.

Network	MIT-States(%)	UT-Zappos(%)
VGG-16	15.4	40.7
VGG-19	15.3	40.8
ResNet-18	17.8	48.3
ResNet-50	19.7	**52.0**
ResNet-101	**20.0**	51.9

Table 2.4: Accuracy of our method with different visual feature extractors.

2.5 Conclusion

In this work, to recognize the unseen attribute-object pair of a given image, we propose an encoder-decoder generative model to bridge visual and linguistic features in a unified end-to-end network. By comparing our method with several state-of-the-art methods on two datasets, we reach the conclusion that 1) the generative model is more competitive than discriminative models to recognize unseen classes, 2) the encoder-decoder mechanism is crucial for learning intrinsic feature representations, and 3) an appropriate model should consider not only the individual property of the attribute and object but also the inner relation between them.

CHAPTER 3

Learning Geometric Fluents Change in Latent Space

3.1 Introduction

With the development of deep Convolution Neural Networks(CNNs), significant progress has been made in recognizing categories of objects in images. However, how to recognize the underlying transformation of objects between two or more images has not been well addressed yet and stays challenging due to big variation of object appearance and noisy background.

Transformations are dynamic changes of object states, which can be visually grouped into changes in geometry and appearance, such as *small* → *big* and *young* → *old* respectively . Understanding transformation is essential for recognizing objects under different states and thus leads to concept formation of objects in semantic perspective. An intuitive way to recognize the transformation of objects is to treat it as a classification problem like traditional object recognition task by manually defining all possible transformations [ILA15, LWZ17]. But the defined categories may lack generality because most transformation are continuous changes. The continuity of transformation space makes it difficult for human to determine the exact degree of the transformations, for example to what extent should we label the door 'open'. Also some transformations described by same semantic word has a big variation in appearance and geometry such as *open book* and *open drawer*.

Inspired by prior works [CDH16, HKW11, KWK15, WGT17, ZB16] which studied transformation with generative networks, We seek to learn transformations in latent space, where images are encoded by low dimensional 'codes'. Transformation function $F(\Theta)$, which describes changes of object between images in real world, is controlled by parameters Θ. It can

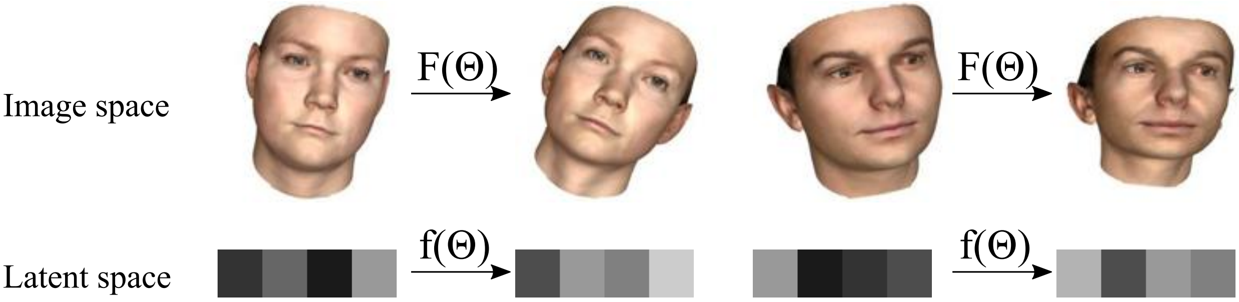


Figure 3.1: Illustration of transformation function $F(\Theta)$ and its mapping function $f(\Theta)$ in latent space. $F(\Theta)$ transforms image in real world with parameters Θ, $f(\Theta)$ transforms image code in latent space with the same parameter. Parameters describing the same transformation are identical in both F and f.

be represented by an unknown function $f(\Theta)$ in latent space, which maps pre-transformation image code to post-transformation image code, as illustrated in Figure 3.1. Instead of trying to explore the transformation function $f(\Theta)$ in given latent space, we define the form of $f(\Theta)$ in latent space explicitly and try to find the latent space where real world transformation function $F(\Theta)$ is mapping to our designed form of $f(\Theta)$.

Following this spirit, we propose a 4-channel Siamese encoder-decoder networks which has three properties listed as below. 1. This networks can predict the representation of transformation of image pair. 2. It can apply desired transformation to the image through explicit operation on image code and transformation code in latent space. 3. The representation of transformation is interpretable [WGT17] which means we can directly design the representation of transformation with unit transformation code.

The transformations we study in this work includes rotation, translation, stretching (scaling along single axis), uniform scaling and face size deformations. These transformations can be further grouped into in-plane transformations and out-of-plane transformations, or linear transformations and non-linear transformation in 3D Euclidean space. The data we use are image pairs rendered from 3d models in graphics engine, with which we can precisely control the parameters of transformations applied to the model.

In summary, our contributions are in three folds:

1) We study the problem of learning transformations between image pairs with weakly-supervised method from only raw image data which is easy to obtain from graphics engine.
2) We have proposed a 4-channel Siamese convolutional encoder-decoder networks which consists of transformation representation extracting networks and transformation applying networks.
3) We have learned an interpretable transformation space in which we can generate desired transformations.

3.2 Related Work

Object state recognition. Object state recognition is closely related to object attribute recognition. People define objects' attributes with several semantic concepts. Each attribute has its own state space, such as {male, female} for *gender*, {open,close} for *door*. Isola *et al.* [ILA15] have collected a large dataset consisting of images described by various adjectives, his learned images representations can be used to distinguish different states of the same attribute independent of objects such as {raw, cooked} and retrieve middle-transformation image. Misra *et al.* [MGH17] have developed a network which combined different object classifiers and attributes classifiers to predict the label of unseen combinations of visual primitive. Liu *et al.* [LWZ17] have jointly modeled objects fluents and action which changes it to improve objects states recognition and action recognition. They all tackle the problem of recognizing object states with a supervised method, although their learned attributes representation can be generalized across different objects, they cannot tackle continuous changes well because of the classification training method.

Encoder-decoder networks. As an unsupervised method, encode-decoder networks have caught a lot of attentions during past years with the development of convolutional networks. They have been applied in many computer vision generation problems with attractive visual results such as different styles image generation [IZZ17], future prediction from time-lapse videos [ZB16], 3D models generated from 2D images [WZX16]. One problem that is close to our works is view synthesis [PYY17, YRY15, ZTS16], they coded transformation parameters and apply it to the latent representation. Our work differs from them in the way that we train the encoder-decoder networks by recovering transformation representation and reconstructing images after transformation from two pairs of images simultaneously without explicit parameter inputs such as angle of azimuth and elevation.

Interpretable transformation networks. There are several exciting works on the study of interpretable representation of transformation between two images or video frames [CDH16, HKW11, KWK15, WCK16, WGT17]. In deep convolutional inverse graphics network(DC-IGN), Kulkarni *et al.* [KWK15] treat transformation learning as an inverse graphics problem, the graphics code is a natural disentangle representation, however the data need supervised organization during training process. Chen *et al.* [CDH16] managed to learn the disentangle transformation code by maximizing mutual information through InfoGan. But their network cannot output a new image which generated from some transformation applied to an input image. Worrall *et al.* [WGT17] study the rigid body transformation through encoder-decoder networks. They have designed a hidden state transformation layer which takes in explicit geometry parameters as inputs besides original images, which generate impressive images.

3.3 Methods

Images are very high dimensional data, whereas the transformations we study can be represented by several parameters, which guides us to seek a relatively low-dimensional space where slightly changing or unchangeable appearance details are highly compressed. Inspired by dimension deduction technology with autoencoder which maps raw data to a latent space through convolutional layers and non-linear activation function such as ReLU, Tanh, we can study the transformation in a low dimensional latent space. However the latent spaces directly learned from neural networks are always hard to interpretable, so we add constraints to train the encoder-decoder which maps data to a latent space with our desired properties.

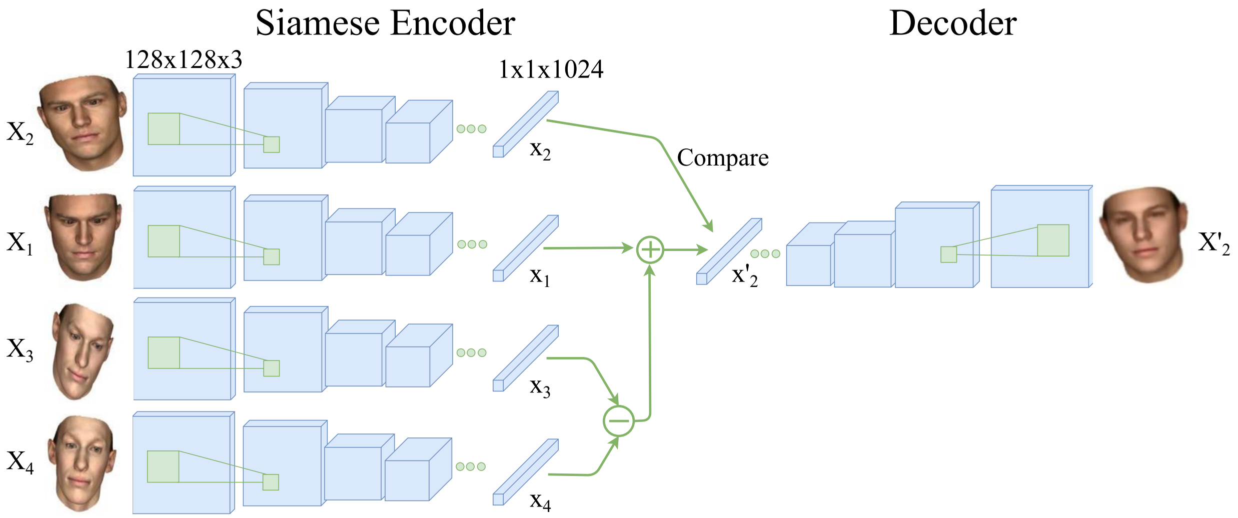


Figure 3.2: Architecture of 4-channel Siamese encoder-decoder networks.

3.3.1 Encoder-decoder networks

Inspired by previous works on learning transformation through auto-encoders styles networks [KWK15, WGT17, ZB16]. We have designed a fully convolutional encoder-decoder networks. Our input images are generated with the size $128 \times 128 \times 3$, the values of each channel are normalize to $[-1, 1]$. Both encoder and decoder are consisting of 7 convolution/deconvolution layers with symmetric channels and filters settings. The filter size for each layer is 4×4, with the strides $(2, 2)$. So the image size is dividing/multiplying by 2 after every convolution/deconvolution layer. The number of channels for each layer are 64,128,256,512,512,1024,1024. Each convolution/deconvolution is followed by batch normalization layer and activation layer except first convolution layer. The activate function for encoder are all LeakyReLUs with slope = 0.2, for decoder we use ReLUs with no slope. We add an extra convolution layer with 3 channels and Tanh activation to make the output fit the size of input images.

3.3.2 Transformations in latent space

Suppose $\mathbf{X}$ is a set of image pairs, the transformation for any given images pair $(X_1, X_2) \in \mathbf{X}$ is controlled by parameter $\Theta = (\theta_1, \theta_2, ..., \theta_n)$, which can be very high-dimensional for complex non-linear transformations. The transformation of (X_1, X_2) in real world can be described by a unified function F:

$$X_2 = F(X_1; \Theta) \tag{3.1}$$

we represent encoder networks and decoder networks as functions $\mathbf{e}$ and $\mathbf{e^{-1}}$ which map between raw image X and its representation x in latent space V. Each encoder networks can map X to its own space V.

$$\mathbf{e}_V(X) = x \tag{3.2}$$

$$\mathbf{e^{-1}}_V(x) \approx X \tag{3.3}$$

Eq 3.3 is not strictly equal due to the information loss in dimension reduction in Eq 3.2.

The transformation function f_V in hidden space V satisfies:

$$\forall (X_1,X_2) \in \mathbf{X}, \Theta \text{ when } X_2 = F(X_1; \Theta)$$
$$f_V(\mathbf{e}(X_1); \Theta) = \mathbf{e}(X_2)$$

There are multiple instances of (V, f_V) because each pair of $\mathbf{e}_V$ and $\mathbf{e}_V^{-1}$ defines a latent space V. In different latent space V, the transformation function f_V can be in very different forms. In this work we study a specific form of f_V, where x and parameters Θ are decoupling by encoders. We call $A(\Theta)$ as transformation vector.

$$f_V(x; \Theta) = x + A(\Theta) \tag{3.4}$$

Suppose we are given two image pairs (X_1, X_2) and (X_3, X_4) which undergoing the same transformation:

$$F(X_1; \Theta) = X_2 \quad F(X_3; \Theta) = X_4$$

In space V, we have:

$$x_i = \mathbf{e}_V(X_i), i = 1, 2, 3, 4$$

$$f_V(x_1; \Theta) = x_1 + A(\Theta) = x_2$$

$$f_V(x_3; \Theta) = x_3 + A(\Theta) = x_4$$

So we have:

$$x_2 = x_1 + A(\Theta) = x_1 + (x_4 - x_3) \tag{3.5}$$

So the transformation vector $A(\Theta)$ can be represented as the subtraction of x_4 and x_3. Below, we will show one interesting properties of $A(\Theta)$ which is similar to linearity of functions.

Transformation decomposition Suppose for three given images, X_1, X_2, X_3, which satisfy:

$$F(X_1; \Theta_1) = X_2$$

$$F(X_2; \Theta_2) = X_3$$

$$F(X_1; \Theta_1 \bigoplus \Theta_2) = X_3$$

$\bigoplus$ is defined as the element-wise summation operator of Θ. In this work, for parameters controlling rotation, translation and face size, it applies the arithmetic summation. For parameters controlling scaling, it applies summation of logarithm of scaling coefficient. In hidden space V, we have:

$$x_2 = x_1 + A(\Theta_1)$$

$$x_3 = x_2 + A(\Theta_2)$$

$$x_3 = x_1 + A(\Theta_1 \bigoplus \Theta_2)$$

From above we can derive that:

$$A(\Theta_1 \bigoplus \Theta_2) = A(\Theta_1) + A(\Theta_2) \tag{3.6}$$

3.3.3 4-channel Siamese encoder-decoder networks

In all, our purpose is to find a hidden space V of which the transformation function f_V has the form as Eq. 3.4.

The latent space V is defined by the encoder networks and decoder networks because encoder does the mapping $\mathbf{e}_V : \mathbf{X} \rightarrow V$, and $\mathbf{e}_V^{-1} : V \rightarrow \mathbf{X}$.

As we show with Eq. 3.5, $A(\Theta)$ can be represented by subtraction of the latent code of images pair. So we design a 4-channel Siamese encoder-decoder networks which is shown in Figure 3.2. The inputs are 2 pairs of images $(X_1, X_2), (X_3, X_4)$, the transformation in each pair are identical. We use a Siamese style networks because $(X_1, X_2), (X_3, X_4)$ should be mapped into the same latent space V by feed them into the same encoder-networks. The 2-channel Siamese transformation generating networks take (X_3, X_4) as inputs, and produce transformation vector $A(\Theta) = x_4 - x_3$. x'_2 is the latent code to reconstruct X_2 where $x'_2 = x_1 + (x_4 - x_3)$. x_2 is obtained for the comparison with x'_2 to calculate reconstruction loss in latent space.

3.3.4 Loss function

To get both accurate pos-transformation prediction in image and latent space, we have introduced two main loss to minimize with back propagation.

3.3.4.1 Image reconstruction loss

The output of our model is the predicted image after transformation. We measure images reconstruction loss with several metrics compared to the ground truth images.

$\mathbf{L_p}$-norm loss. A typical loss is pixel-wise Mean Square Error(mse), also known as L2 loss. However it is only good at capturing low frequency information, the image generated with only L2 loss is always blurred. We adopt L1 loss in our work which causes less blurring meanwhile can still capture accurate low frequencies as suggested in [IZZ17].

Gradient difference loss. As suggested in [MCL15], Gradient Difference can capture high frequencies such as edges and corners because it calculates the gradient between neighboring pixels. Lowing Gradient Difference Loss(GDL) can sharpen the generated image. The GDL loss between ground truth image X and predicted image $\hat{X}$ is calculated as:

$$\begin{aligned} L_{gdl} = \sum_{i,j} & ||X_{i,j} - X_{i-1,j}| - |\hat{X}_{i,j} - \hat{X}_{i-1,j}||^2 + \\ & ||X_{i,j} - X_{i,j-1}| - |\hat{X}_{i,j} - \hat{X}_{i,j-1}||^2 \end{aligned} \tag{3.7}$$

Structural similarity loss. Structural similarity index(SSIM) [WBS04], which is also widely used, has demonstrated its effectiveness for capturing structural information such as luminance, contrast. We use structural dissimilarity(DSSIM) as structural loss function to minimize:

$$L_{dssim} = \frac{1 - SSIM(X, \hat{X})}{2} \tag{3.8}$$

Our whole image reconstruction loss is defined as the weighted combination of the above three losses.

$$\mathcal{L}_{img} = \lambda_{L_1}\mathcal{L}_1 + \lambda_{gdl}\mathcal{L}_{gdl} + \lambda_{dssim}\mathcal{L}_{dssim} \tag{3.9}$$

3.3.4.2 Code reconstruction loss

We not only want to minimize the loss from ground truth image and generated image, but also we design a loss to penalize the difference between ground truth image code and reconstruction code in latent space V, which we call code reconstruction loss. The purpose that we introduce this loss is to encourage our learned decoder networks mapping the same space V as encoder networks mapping to the image space. We use $\mathcal{L}_2$ loss to measure the difference between two vectors.

$$\mathcal{L}_{code} = \mathcal{L}_2(f(\mathbf{e}(X_1); \Theta), \mathbf{e}(X_2)) \tag{3.10}$$

So our final loss function is:

$$\mathcal{L}_{rec} = \mathcal{L}_{img} + \lambda_{L_{code}}\mathcal{L}_{code} \tag{3.11}$$

3.4 Experiments

Below we will show our settings for training, datasets, visual results and analysis.

3.4.1 Training settings

In training phase, we minimize our loss function with Adam Stochastic Optimization [KB14] with initial learning rate $lr = 0.0001$, the mini-batch size is 64. The model is training with 40K iterations. We implemented our whole model in Keras [Cho15] environment, with TensorFlow as back-end. We set $[1, 2, 1, 1000]$ for hyper-parameters $[\lambda_{L_1}, \lambda_{gdl}, \lambda_{dssim}, \lambda_{code}]$.

3.4.2 Faces dataset

Basel Face dataset [PKA09] is a PCA based face model from which we can randomly generate 3D faces. We rendered the images from 3D models with perspective projection methods to make the images more realistic. We augmented the face model with 4 types of transformations: scalings, rotations, translations and face size deformations and their combinations. For scalings we generate separately for x-axis, y-axis, and uniform-scaling. For rotations we generate 2D rotations(roll), 3D rotations(change azimuth and elevation). For translation, we only perform in-plane translation, as out-of-plane translation is visually same as uniform scaling from perspective view. For face size deformation, besides their main PCA model which is used to generated faces, they offer another attributes PCA model through which we can generate faces with different sizes (fatter or thinner). All rendered images are re-sized to 128×128 with 3 color channels.

For one sample of transformation in our training sets, we generate 4 images (a_1,a_2,b_1,b_2), where the transformations of $a_1 \rightarrow a_2$ and $b_1 \rightarrow b_2$ are identical but a_1 and b_1 have different initial weights in PCA components, positions and poses. Since all the transformations we studied can be inverted, so we can further generate 3 training samples by simply re-ordering the elements of the original tuple to get (b_1,b_2,a_1,a_2),(a_2,a_1,b_2,b_1), (b_2,b_1,a_2,a_1). We have generated around 108K 4-elements tuple in total and split them into training and validation sets with the ratio 8:2. An illustration of training sample is shown in Figure 3.3.

3.4.3 Image generation results

3.4.3.1 Image sequence generation

To show the generative ability of our trained networks, we generate face images by varying only one transformation parameter for better visualization and comparison. Faces in the

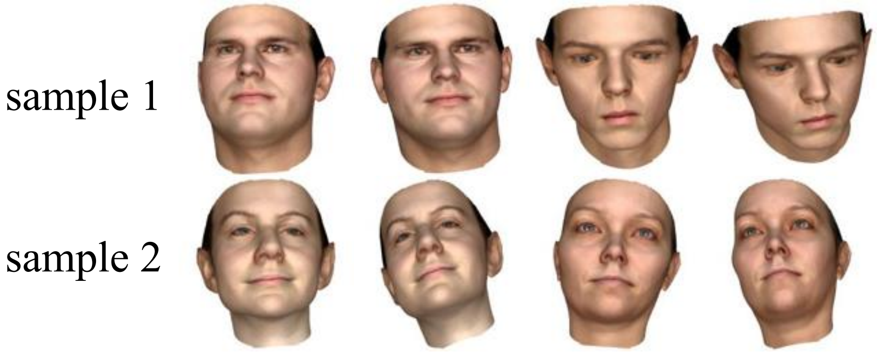


Figure 3.3: Illustration of training samples.

center of every row are the 'starting' faces. First row and second row are showing reference faces and ground truth faces respectively. Note that they are generated with different initial parameters. The parameter intervals between every two adjacent columns are identical. The range of each parameter, azimuth: [-24°, 24°], elevation: [-24°, 24°], roll:[-45°, 45°], y-axis translation:[-20px, 20px], x-axis scaling: [0.96^4, 0.96^{-4}], size: [-80, 80].

We have used 3 methods to generated the faces. **Directly mapping:** This is exact the same way how we trained our networks, our model predicts the faces based on two reference faces in the first row. **Image by image:** We first generate images in 4-th and 6-th columns with directly mapping. Then we generate other columns recursively based on former generated image. For example we generate images in 3-rd column based on 4-th and 5-th faces for transformation reference in first row and the faces we just generated in 4-th column. **Code by code:** Rather than generating images by images, we extract the transformation vector between 5-th image and 6-th image in first row as unit transformation vector. Then we recursively generate images in the other columns by keeping adding unit transformation vector or subtracting it.

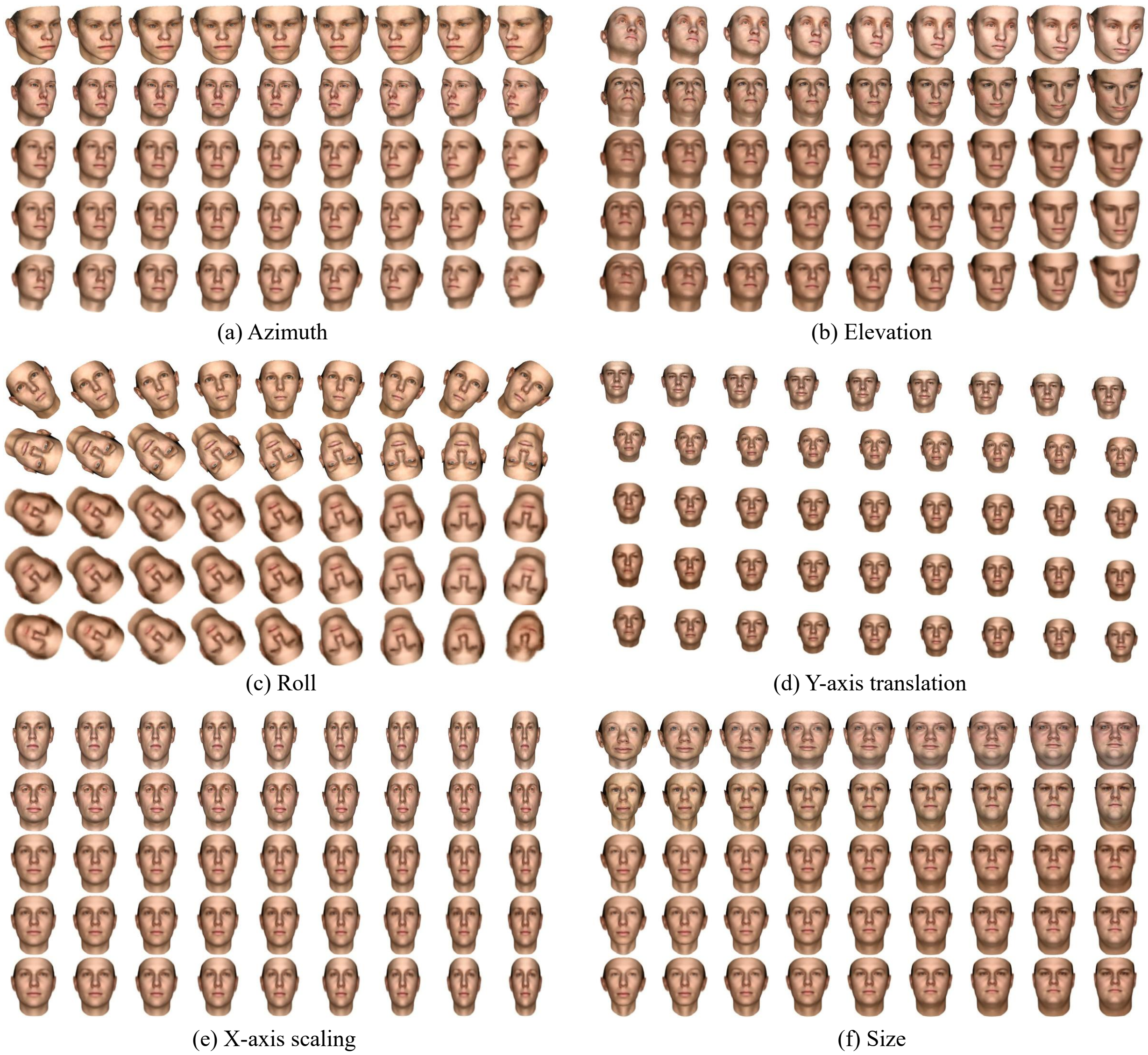
The results are shown in Figure 3.4. From the results we can see all of the three methods are giving visually satisfying results, despite of some details of faces cannot be recovered. Besides, **image by image** method has shown that our learned transformations are independent of the absolute poses and positions of two faces, which can truly represent the relative change between two images. **Code by code** method further demonstrates that our learned transformation space has the property of transformation decomposition and combination just as we derive in Eq. 3.6, based on which we can generate more faces.

3.4.3.2 Transformation vector generation

As what we have shown in Eq. 3.6 and our **image by image** and **code by code** methods demonstrates, we can precisely generate transformations by just generating transformation vectors in hidden space instead of generating images pair which undergoing transformations as our transformation reference. For azimuth, elevation, face size and x-axis scaling, we generate 10 images pairs undergoing a certain degree of transformation for each, then we extract their transformation vectors and average for each group to reduce variance. We treat these four 1024-dim vectors as our unit transformation vectors. We randomly generate 9 pairs of images undergoing the transformation controlled only by azimuth angle, elevation angle, face size coefficient and x-axis scaling coefficient. Then we generate transformation vectors with arithmetic combination of our four learned unit transformation vectors, our final faces generation results are shown in Figure 3.5. In spite of lack of some facial details, the pose, position and shape of the generated faces look no different, which are crucial to determine the transformations. These results demonstrate the interpretability of our learned model, and we can generate faces with the latent code of unit transformation vectors.

3.4.4 Visualization of transformation space

In order to have a more straightforward visualization of the transformation space we have learned. We have extracted all of the 1024-dimensional transformation vectors from our testing sets. Since our transformations has a natural form indicating that it can be decomposed as the summation of multiple sub-transformations, which is very similar to PCA. We do the PCA analysis of our extracted transformation vectors and choose top 20 components, so the transformation vectors are further deducted to 20 dimensions. We visualize it with t-SNE [MH08], we set perplexity = 1000 since our dataset is large and we want to observe the global geometry as suggested in [WVJ16]. The results are shown in Figure 3.6.



(a) Azimuth

(b) Elevation

(c) Roll

(d) Y-axis translation

(e) X-axis scaling

(f) Size

Figure 3.4: Faces generation results with one parameter varied. For each subplot, Row 1: Reference images for transformation; Row 2: Ground truth images; Row 3: Images generated with **directly mapping** method; Row 4: Images generated with **image by image** method; Row 5: Images generated with **code by code** method

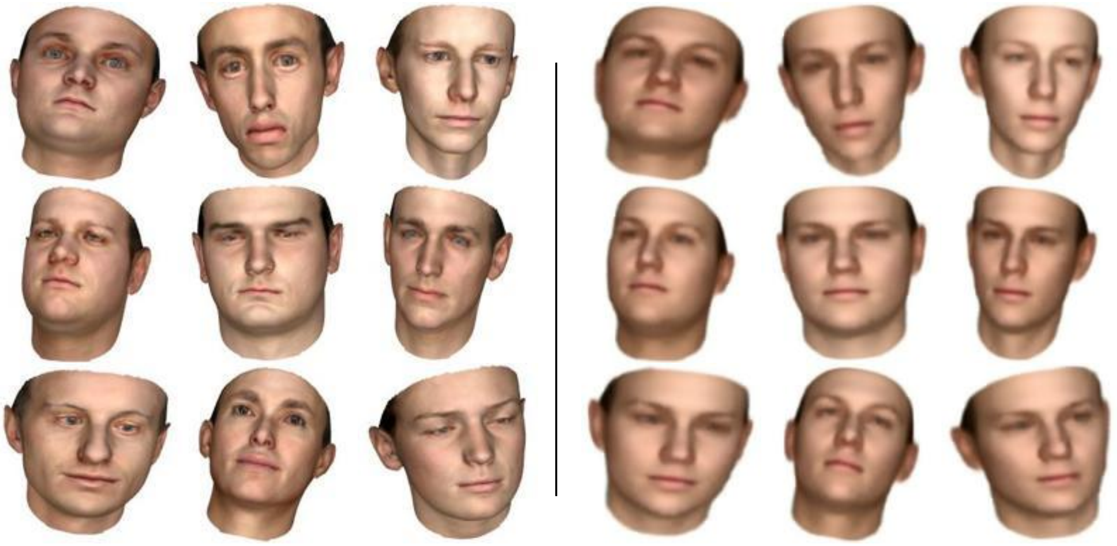


Figure 3.5: Illustration of faces generation results with transformation vectors constructed with unit transformation vectors. Faces on the left are ground truth images. On the right are our generated faces undergoing desired transformations.

The visualization looks very reasonable. First we can see transformations that only varies one parameter cluster together and form two stripes, each of them represents the positive or negative direction of parameters change. However the in-plane rotation tends to form a circle because of its periodicity. Second, to check if our space is interpretable, which means ordered transformations in real world are also ordered in our transformation space. We select transformation vectors in 3 transformation categories: azimuth, elevation and face size, and then apply them to a new generated face. The transformation vectors we selected are plotted as red dots in Figure 3.6. As the absolute value of parameter increases, the dot tends to get further from original point. We have put some generated face next to the dots which represent their transformations from the original face at original point.

3.5 Conclusion

In this work, we study the problem of learning transformation space in 2D images. The transformations we are studying includes both in-plane, out-of-plane, linear and non-linear

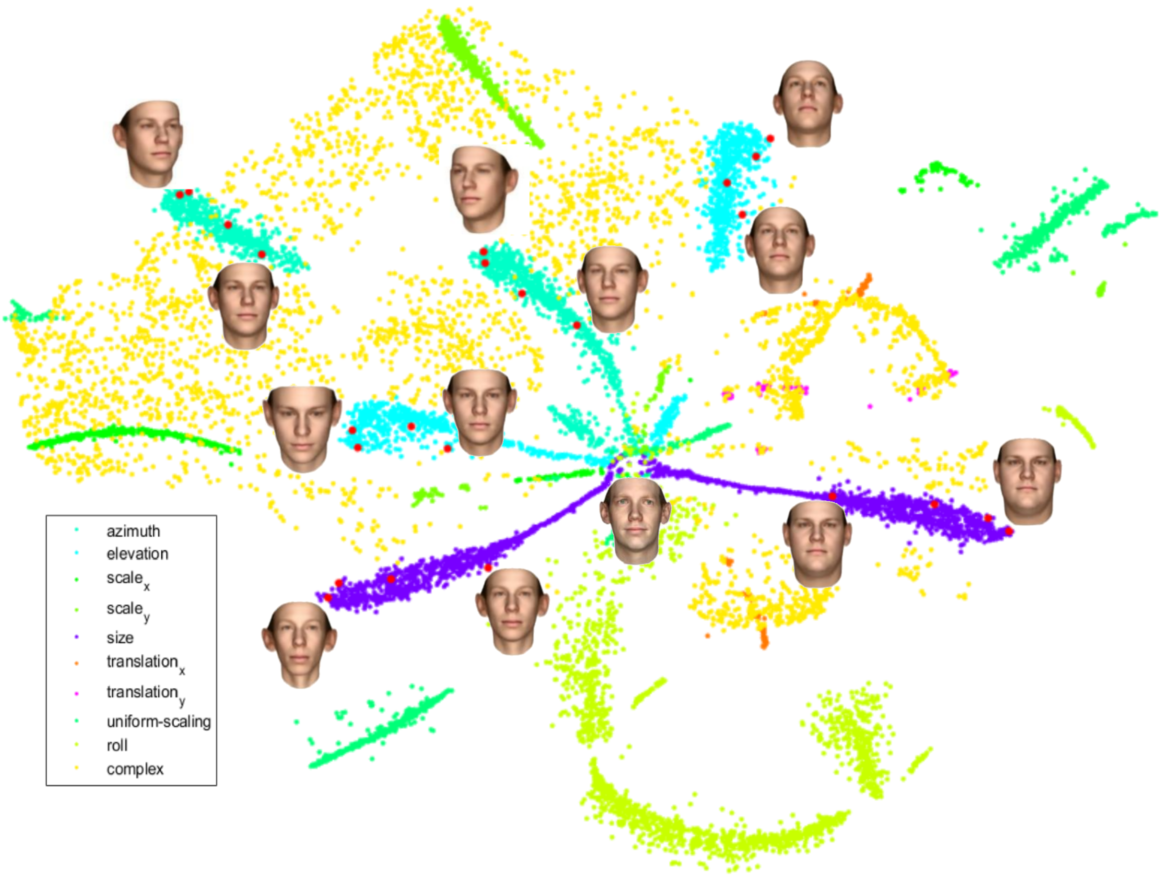


Figure 3.6: t-SNE visualization of our learned transformation space. Each color represents a certain transformation with only one parameter varied, while dots in orange mean the complex transformations consisted of multiple parameters varied. The face at the junction is the original face, the other faces at different places mean they are generated with the transformation vectors(red dot) next to them.

transformations. We have proposed a 4-channel Siamese encoder-decoder networks which learns transformation space with 2 pairs of images undergoing identical transformations. We show the strength of our model by demonstrating our generated images with three different generating methods and the visualization of learned transformation space. For future works, we will discover transformation function in more complex forms in latent space and study more complex transformations in real scenes.

CHAPTER 4

Learning Interpretable Fluents Change and Intuitive Physics

4.1 Introduction

In last chapter we have discussed how to learn a set of implicit geometric fluents change by providing designed quadruple to enforcing the linearity of parameters in latent space. However it is hard to interpret the meaning of each dimension of the latent vector, and in most of the case some single fluents such as rotation angle, scaling coefficient are represented by multiple dimensions. In this chapter we are seeking a more natural way to learn the latent fluents space with more meaningful dimensions in physics.

When humans in their infancy, they developed quickly how to perceive the world and predict some simple physics phenomenon which is the so-called intuitive physics [SC94]. What will a physics system, such as a number of balls in a box, look like in 10 seconds? For us, we may tell a short future of this system by our feeling. However for a physician, if the system is ideal and the physical property for each entity are known, then the dynamics of them can be calculated accurately although might be hard. The calculation for dynamics is based on the well-known physics law like Newton's law of motion. In this chapter, we are interested in letting the computers make precise prediction of some physics phenomenon like collision and gravity.

Just like how physicians proposed the hypothesis of law from daily observations and verified it with designed experiments. In this work, we are trying to use computer vision technique to learn the intuitive physics by observing the videos, and apply the law we have learnt in order to predict future in an unsupervised manner. The future of physics phenomenon is affected by lots of physical properties like the touching point, elastic coefficient, masses and shapes of the objects. It is expected that the latent space we learn will be capable to have all these information encoded in order to learn a meaningful physics law represented by the dimension in fluents space.

Following the procedure of solving a physics problem: 1. Finding interested objects 2. Measuring physics properties for those objects 3. Calculating the future dynamics according to the physics laws. We build up our system with segmentation module, physics state encoder, physics state predictor respectively. We have further designed an rendering module for purpose of calculating loss on raw images and visualization.

We have conducted our experiments on our newly collected dataset generated with a graphics engine called Box2D with controllable parameters for each entity. Note that the simulation in the graphics engine are based on the pre-defined rule, so what we are trying to learn are the graphics rules in the simulation engine, making our problem as an inverse graphics problem.

4.2 Related Work

Video prediction. With the development of generative networks such as autoencoder and GAN, video prediction or interpolation have been studied intensively recently [MCL15, FGL16]. Those works are focusing mainly on the motion information in the past frames like velocity and acceleration or try to memorize a certain type of transformation [ZB16]. Although our work is a kind of future prediction, it is different from them in the aspect that our model not only perceive motion information, but our model can also predict the future based on the other perceived properties and the rule learnt from the system. Our system doesn't simply predict the future with linear extrapolation/interpolation of velocity or acceleration, instead we predict the future position of objects based on the object states and system rule. With this rule we can predict the future for a relatively large time span compared to traditional video prediction which can only deal with simple motion.

Inverse Graphics. Graphics engine produce the images with 2 stages: 1. Accept the parameters that control the properties of objects such as color, shape, size 2. Render the images with the controlling parameters. While computer vision in the inverse process of graphics, by processing the images, computers can infer the parameters which is also called representation learning. With the development of deep neural networks, a conventional model for inverse graphics problem is autoencoder [KWK15]. The encoder networks serve the role of perceiving module while the decoder networks can be seen as the rendering module. The graphics code to learn are naturally lying in the latent space. In our works, to predict future frames of objects in the physics phenomenon is equal to predict the graphics code and input them to rendering engine.

Intuitive Physics. In real world, people can quickly judge if two objects will collide or not by estimate their velocity, position and their friction with the environment. Works on learning intuitive physics are mainly on three aspects. The first setting is with explicit parameters and certain expression of equations such as polynomial, this is mainly a regression problem in statistics. However the form of expression may be very complex or unknown, hence they are approximate with probabilistic models [WYL15]. The most general setting is with implicit parameters and expression forms, several works such as block towers [LGF16, WLK17] and object collisions [BPL16, WZW17].

4.3 Dataset

In this work, we have collected a large dataset with an open source physics simulation engine called Box2D. We simulate our data by placing multiple objects in a close box and giving each object a random initial velocity. We collected the rendering images, and physical properties such as mass, position and velocity for 40 frames in one run. The controlling parameters which can be initialized randomly are numbers of objects, object size, position, velocity and color. The physics interaction types we simulate are collision and n bodies gravity system. A snippet of our data samples can be viewed in Figure 4.1.

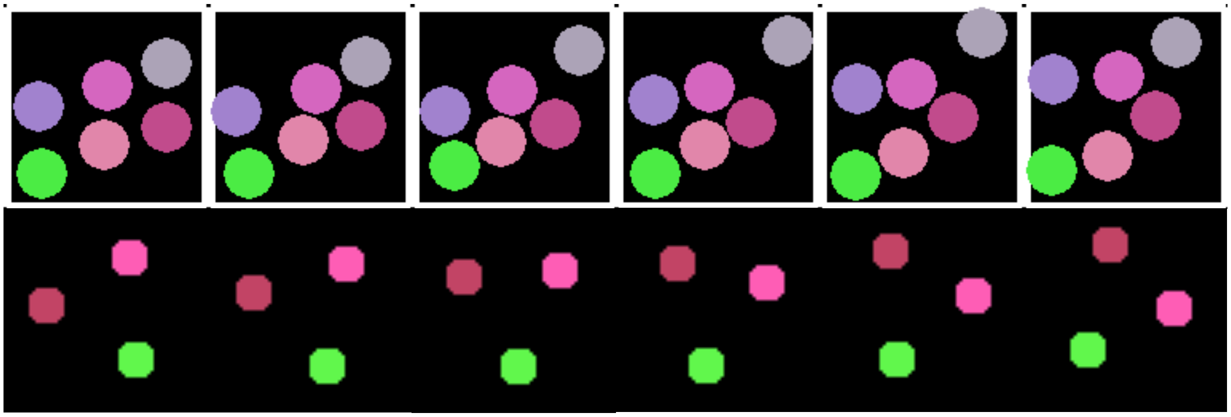


Figure 4.1: Each row contains the 1st, 4th, 7th, 10th, 13th, 16th frame of the corresponding video.

4.4 Approach

To predict the future video frames, we design the network architecture following two steps: 1. Design the autoencoder networks which can encoder the video frame to latent vector and also render it with the decoder. 2. Design the physics inference engine with can predict the latent vector.

4.4.1 Autoencoder networks

First we will introduce the autoencoder networks. For the purpose of simplification, we assume the segmentation of each object is available to us. For each object, we use the identical autoencoder networks to transform the image with the object segmented out to a low dimensional vector in latent space.

In this work, we have designed two branches for the encoder networks as illustrated in Figure 4.2, which means the latent representation for each object is composited of two latent vectors from two different branches. One part corresponds to variable fluents in the scenarios of object interactions, such as position, orientation, typically the dimension of latent vector for this branch can be very low. The other part corresponds to all the invariable fluents of the objects including size and color.

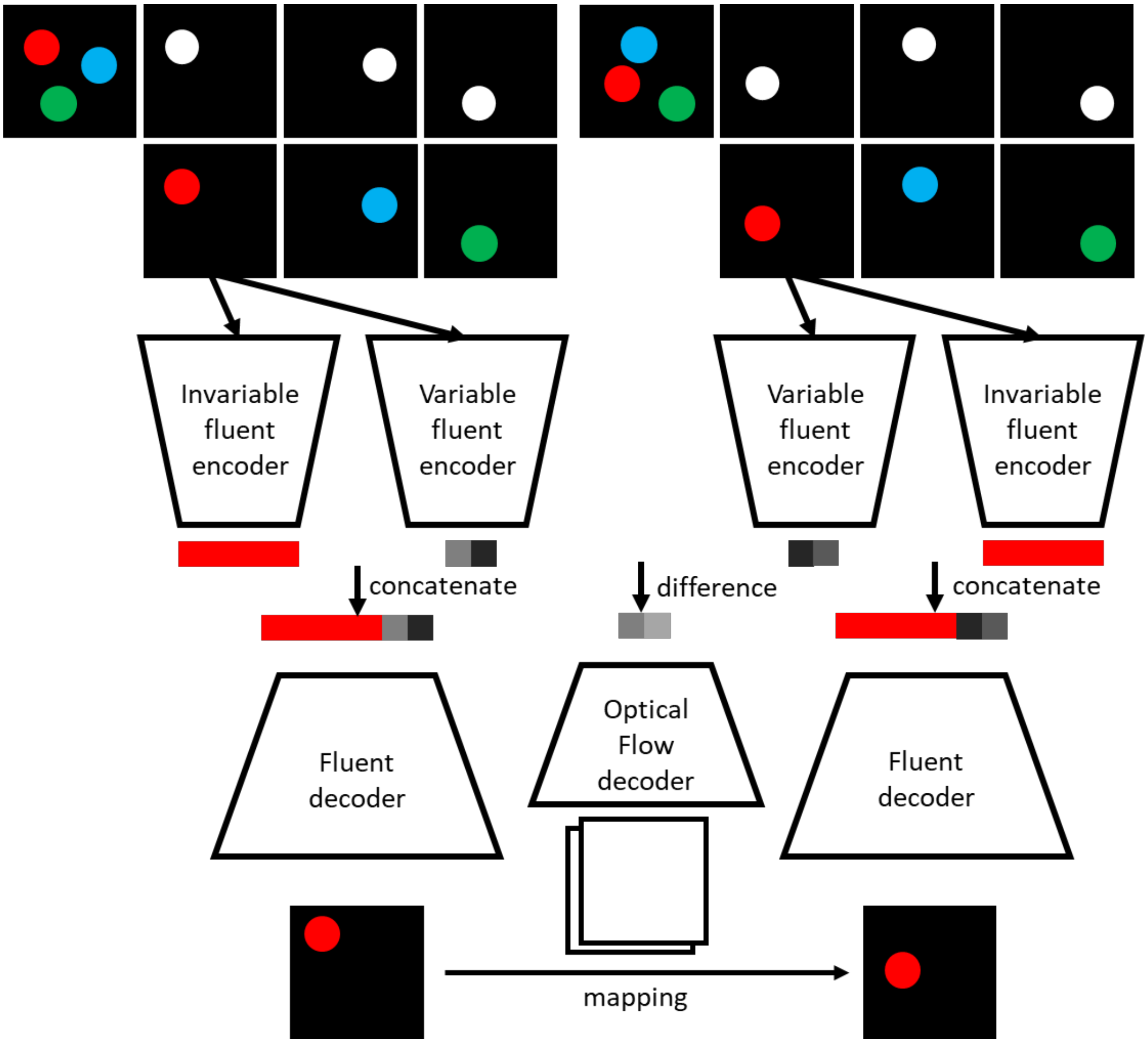


Figure 4.2: Illustration of autoencoder networks.

For the decoder networks, we utilize a 6-layer de-convolution networks to decode the object vector which is basically the concatenation of two latent vectors from the encoder networks. In addition, we hope the latent space for the variable fluents to share some similar properties with those in image space like linearity, continuity. With the variable fluents space with these kind of properties, the physics dynamics we learn in this latent space will have a similar manifold with that in Cartesian coordinate system of image space. With the above encoder-decoder networks, we can obtain the point to point mapping between image space and latent space. Now we need a vector to vector mapping between these two spaces to

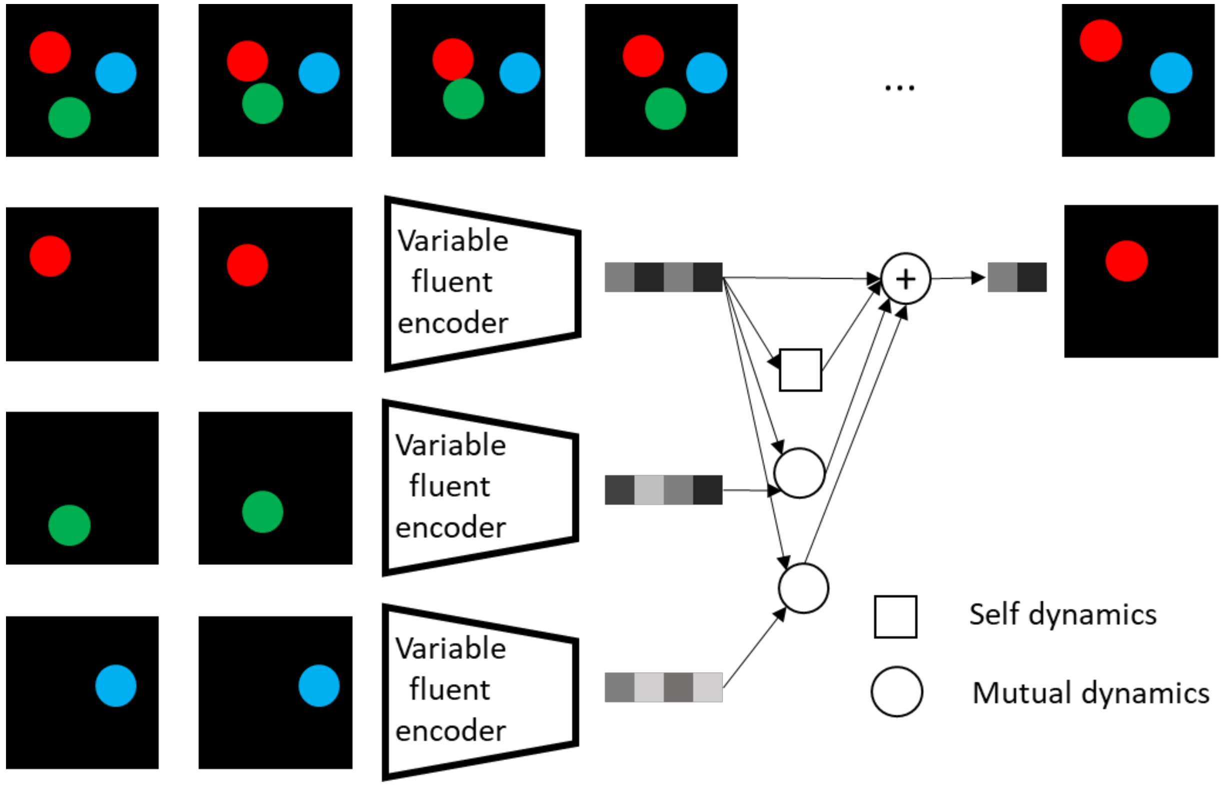


Figure 4.3: Illustration for physics dynamics inference networks.

ensure some more properties. So we have designed an extra branch to decode the difference between two variable vectors which are of the same object in two different frames to the optical flow in the image space which describes the movement of this object in two frames.

4.4.2 Physics dynamics engine

Given first few frames of a video, we want to predict the position and orientation of the objects in future frames. Since this problem is not simply a video frame extrapolation. We have to learn the physics dynamics like law of collision to precisely infer the future states of the objects.

Follow the spirit of Newton's law of motion.

$$\begin{aligned}\Delta x =& x_2 - x_1 \\ =& (v_1 + v_2) \times \Delta t/2 \\ =& v_1 \times \Delta t + 1/2 a \Delta t^2 \\ =& k_1 v_1 + k_2 \Sigma F\end{aligned}$$

The variable fluent vector contains the information of position, then we can approximate velocity from the difference of two variable fluent vectors in consecutive frames. When the object is moving, it will subject to the force from interaction with other objects and environment. Hence for each object, we have designed mutual dynamics module to predict the force brought by the other objects the environment. We add all the forces together to predict the final variable fluent vector for this object in the next frame.

4.5 Experiment Results

In this section, we will describe the details in training. Our training procedure contains two stages.

We first train the invariable and variable autoencoders with three randomly picked images I_1, I_2, I_3 from the same sequences. We can get the invariable vectors for each objects in I_1, and variable vectors for each objects in I_2 and I_3. We have two ways to reconstruct I_3: I_3' is constructed by decoding from the concatenation of invariable vector from I_1 and variable vector from I_3; I_3'' is constructed by first getting the optical follow map from the difference of variable vectors from I_2 and I_3, and then mapping I_2 based on the optical flow map. The loss function to train autoencoders is as follow:

$$L_{autoencoder} = \|I_3 - I_3'\|_2 + \|I_3 - I_3''\|_2$$

Once the model converges in stage 1, we use the parameters in stage 1 as pre-trained weights. We use the latent vectors in two consecutive frames to reconstruct the variable vector in the third frame, and then decode the images with latent vectors we get. The loss is summation of MSE loss over all predicted frames.

4.5.1 Autoencoder

In order to evaluate the variable space we have learnt, we firstly generate invariable vector by sampling from the normal distribution used in variational autoencoder. Then we vary the value for each dimension in variable vector from -1.5 to 1.5 with 0.5 as interval. There are 49 decoding images in total and we have plotted them in Figure 4.4. We can observe that the learnt dimension has some semantic meaning. When varying the value in one dimension, the position of object is also shifting in one direction. More precisely in the figure, the two dimensions in the variable fluent space are aligned with the diagonal and sub-diagonal of the image space, which means the latent space we have learnt can transform to the image space by just rotating about 45 degrees as shown in Figure 4.5.

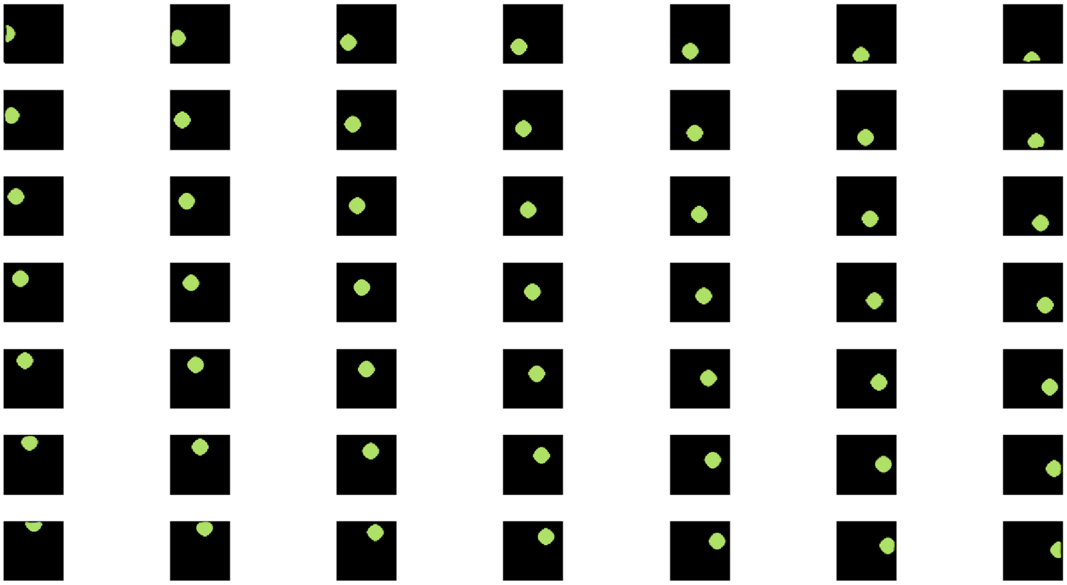


Figure 4.4: This figure shows about the rendering objects by varying value linearly in each dimension of variant fluent space. The varying range for both dimensions are $[-1.5, 1.5]$ with interval of 0.5.

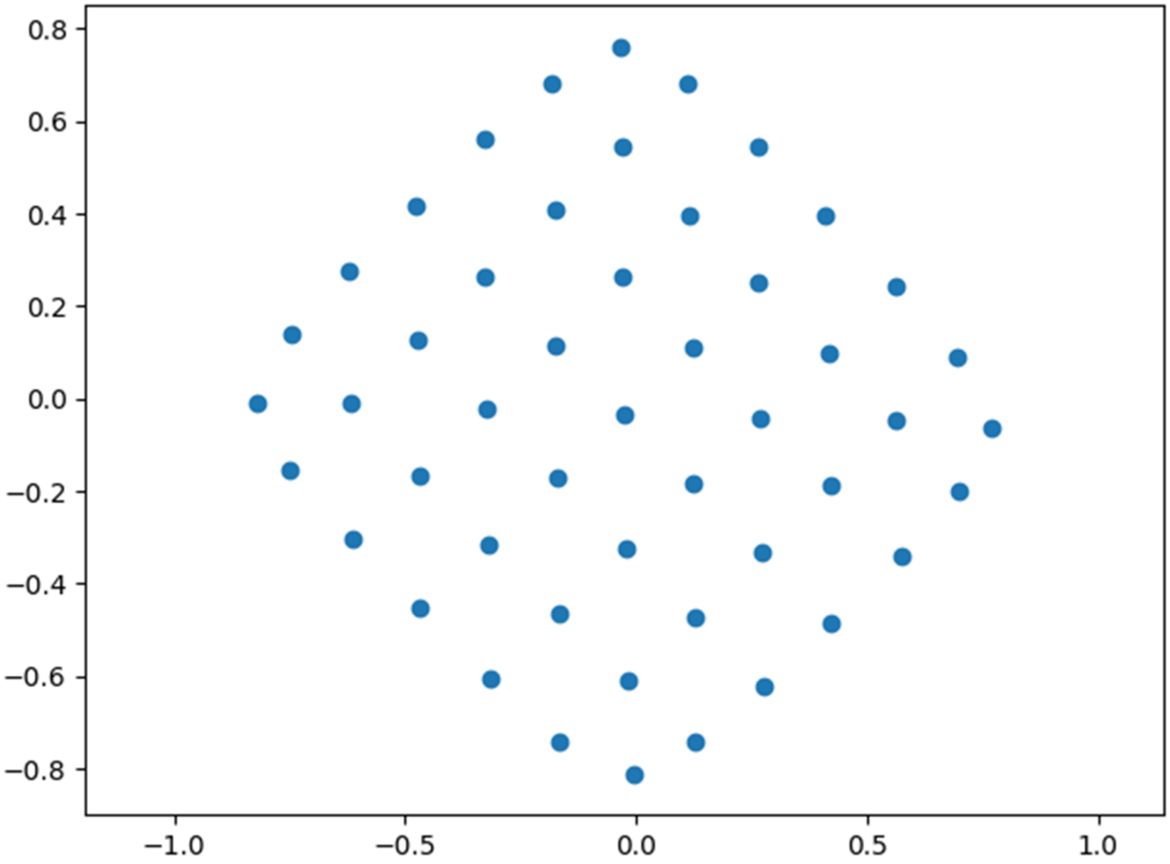


Figure 4.5: This figures shows the coordinates projection from latent space to image space. We can observe that the latent space is nearly uniform which can transform to image space by merely affine transformation.

4.5.2 Physics dynamics

Given a sequence of frames, we initialize our system with first 3 frames of the video. We use the variant encoder to extract variable vector for each object in the first three frames. Then we rollout the variable vector one by one and use then to predict the future frames. The results are shown in Figure 4.6 and Figure 4.7, for N bodies problem, the predicted results are much more close to the ground truth while the results for collision are less precisely. This is because the physics system for collision is more complex than the N bodies problem as it involves sudden interaction between objects, while in N bodies problem all the objects are governed by an unified gravity law across time. However, our model can still generate some qualitative results.

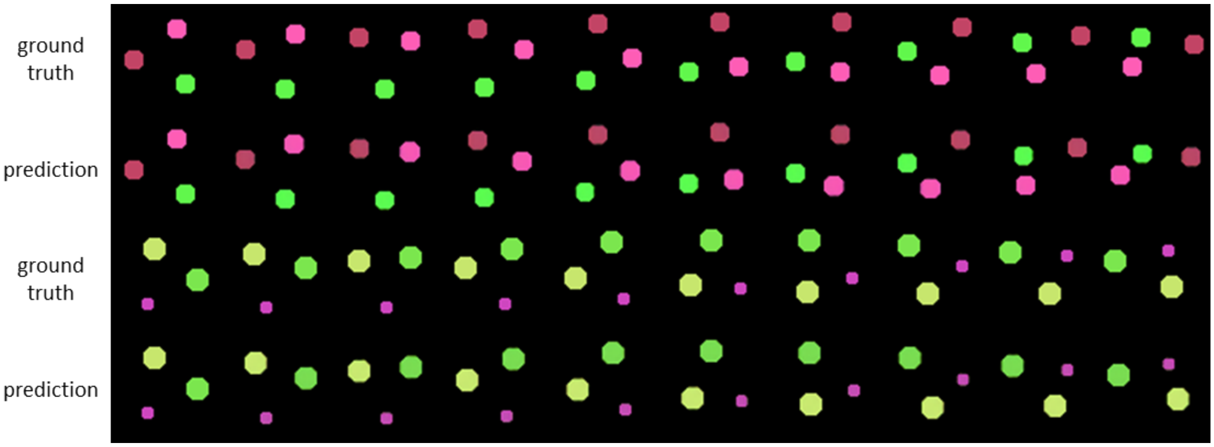


Figure 4.6: Images prediction results for three bodies problem.

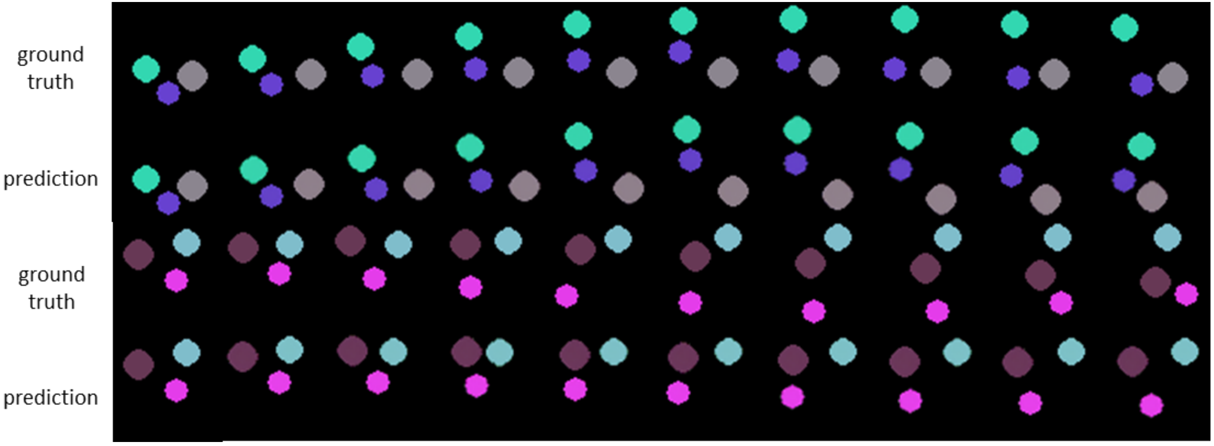


Figure 4.7: Images prediction results for three balls collision.

4.6 Conclusion

In this work, we have seek the possibilities to learn a more disentangled fluent space with each dimension has some semantic meaning. With the simulating dataset we have collected and the model we have learnt, it is showing that understanding inner fluent of object in the image with autoencoder is promising. However, there is still a long way to go and lots of things to explore to rich the fluents space and apply them to more various objects and fluents change.

CHAPTER 5

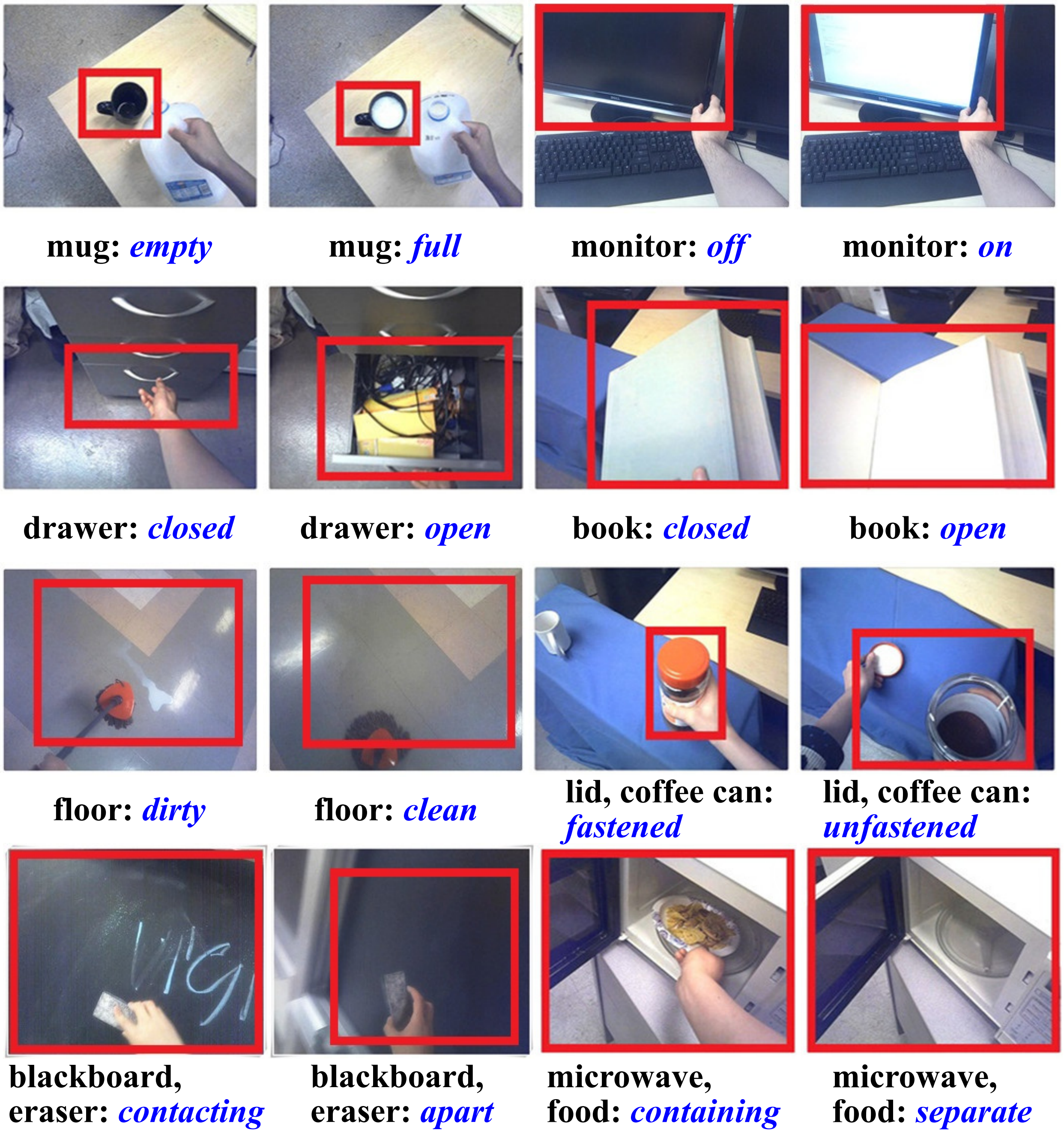
Jointly Recognizing Object Fluents and Tasks in Egocentric Videos

5.1 Introduction

Egocentric vision has attracted a growing attention with the advance of wearable camera technologies, such as smart glasses and virtual reality headsets. The wearable cameras mounted on the head enable a user to record the videos from the first-person view while performing daily tasks.

Two significant issues related to egocentric vision are recognizing tasks and recognizing object fluents. A task is a goal-oriented human activity which interacts with the objects in an environment and changes some attributes of the objects, such as *mop floor*, *make coffee*, and *microwave food.* A fluent is a time-varying attribute of an object or a group of objects, and its values are the specific states of the attribute [FZ15, Mue15], as shown in Figure 5.1. For example, a floor's fluent takes the values *dirty* and *clean* over time as the floor is mopped. In our work, fluents are divided into self fluents and mutual fluents. A self fluent describes the attribute of a single object, such as *dirty* to the floor and *full* to the mug. A mutual fluent describes the attribute of two objects as a whole, such as *fastened* to a lid and a coffee can, *contacting* to a blackboard and an eraser, etc.

One of the most widely-used cues for activity recognition in egocentric videos is the appearance information of related objects [FFR11, FLR12, PR12]. However, in complex goal-oriented tasks, the appearance of the same object often dramatically changes in different phases of the tasks, which may mislead object appearance based activity recognition. For



mug: ***empty***

mug: ***full***

monitor: ***off***

monitor: ***on***

drawer: ***closed***

drawer: ***open***

book: ***closed***

book: ***open***

floor: ***dirty***

floor: ***clean***

lid, coffee can: ***fastened***

lid, coffee can: ***unfastened***

blackboard, eraser: ***contacting***

blackboard, eraser: ***apart***

microwave, food: ***containing***

microwave, food: ***separate***

Figure 5.1: Illustration of self and mutual object fluents. The blue italic words describe the object fluents.

example in Figure 5.1, the appearance of the drawer is vastly different before and after the drawer is opened. This phenomenon motivates us to explore new methods to model and recognize tasks.

We propose to model and understand tasks in egocentric videos from a new perspective - the effects that a task causes. A task is a human activity which aims to change some attributes of the objects in an environment. The accomplishment of a task indicates the realization of one or multiple desired fluent changes, which we call the key fluent changes. For example, the task *sweep floor* changes the floor from *dirty* to *clean*, as shown in Figure 5.2. With the knowledge that the floor becomes clean from being dirty with some stains, we can reasonably infer that the task *mop floor* might have occurred, even if we did not observe any human activity features. In addition to the fluent of floor, the task *sweep floor* also contains several other key fluents, such as *apart* or *contacting* with respect to *broom* and *trash*, *separate* or *containing* with respect to *dustpan* and *trash*. All these fluents contribute to define and discriminate the task *sweep floor*.

Furthermore, different fluents interact closely with each other both in spatial and temporal domains. In spatial domain, the interaction is presented as fluent concurrence, which means some states of two or more different fluents often occur together. For example, in the task *sweep floor*, a *dirty* floor often means the trash is not contained in the dustpan. The fluent *dirty* with respect to *floor* and the fluent *separate* (not contained) with respect to *dustpan* and *trash* occur together. In temporal domain, the interaction is presented as fluent transition, which means fluents in different tasks transition with different probabilities. For example, in the task *write on blackboard*, it is likely that the blackboard changes from *clean* to *dirty*, but unlikely to change in the opposite direction. This case is just the opposite in the task *clean blackboard*.

In this work we propose a fluent-based task representation method to jointly recognize object fluents and tasks in egocentric videos. A task is represented as several key object fluents, which interact with each other by means of concurrence in spatial domain and transition in temporal domain. The task, object fluents, video frames, and the relations among them are described with a unified hierarchical graph. Given an egocentric video, a beam search algorithm [WZZ17] is adopted to jointly infer the object fluents in each video frame and recognize the task of entire sequence. To evaluate the proposed method, we collected a large scale egocentric video dataset of tasks and fluents in daily activity scenes. The experimental results on this dataset show the effectiveness of our method.

This work makes three major contributions:

1) We represent tasks in egocentric videos from a new perspective - representing tasks with object fluents.
2) We propose a hierarchical model to represent the task, object fluents, and their interaction relations in a unified framework.
3) We collected a new egocentric video dataset of tasks and object fluents. The experiments on this dataset prove the strength of our method.

5.2 Related Work

Activity modeling and recognition. Human activity recognition is a classic problem in computer vision and has been intensively studied for decades. Some early studies describe appearance and motion information in 2D images or videos with hand-drafted spatio-temporal features [Lap05, LMS08, SLC04, WKS11, WS13]. Recently, deep learned features from neural networks [KTS14, SZ14a, WQT15] have been applied for activity modeling and produce impressive results. With the advance of motion and depth capture technology, such as Kinect [SFC11], many studies model and analyze human activities in 3D space or RGBD data [KGS13, WZZ17, WZZ13b].

To understand the inner contents of human activity videos, some studies model human activities with hierarchical structures [BT11, RA09, SDN08, SPY11, WZZ17, XLL16]. Wei *et al.* [WZZ17] proposed a 4D human-object interaction model to jointly recognize human activities and localize objects, and they utilized a dynamic beam search algorithm to solve the inference problem in the hierarchical graph. These hierarchical methods inspire us to represent tasks and fluents in a hierarchical structure.

Different from recognizing the activity of the entire video sequence, some studies recognize activities with partial observation or make early detection [HT14, MSS16, Ryo11]. In traditional activity recognition methods, classifiers are trained by encoding the information of the whole video, which is not optimal for activity action recognition with partial observation. Ryoo [Ryo11] utilized sequential matching to early recognize human activities with dynamic bag-of-words. Hoai *et al.* [HT14] used Structured SVM to learn a max-margin early event detector. With the development of deep learning, Recurrent Neural Networks (RNN) such as Long Short Term Memory (LSTM) [Gra12] have been applied to early detection of activities. Ma *et al.* [MSS16] employed LSTM with a designed ranking loss for early activity detection.

Activity recognition in egocentric video. Egocentric video analysis has been paid growing attention with the prevalence of egocentric cameras [FFR11, FLR12, LYR15, MFK16, PR12, SAJ16]. Activity recognition becomes more challenging in egocentric videos since the movement of camera may lower the performance of the traditional hand-drafted spatial-temporal features such as STIP [Lap05], Dense Trajectory [WKS11], and some deep learned features [SZ14a]. Moreover, the human body features become weak or even invisible in egocentric videos. To overcome these difficulties, several semantic egocentric cues have been discovered, such as object cues [FFR11, FLR12, PR12], gaze cues obtained by eye-tracking glasses [FLR12, LYR15], and hand cues [LYR15, MFK16, SAJ16]. Li *et al.* [LYR15] elaborately evaluate various mid-level egocentric cues for action recognition and achieved impressive results with different combinations of those cues. Inspired by previous works, descriptors extracted from multi-stream networks [MFK16, SAJ16] encoding different cues have proved efficient.

Object states and fluents. Object fluents are used to describe object states [FZ15, Mue15]. Object state detection and recognition has been recently studied in still images [DPC12, ILA15, ZDG14]. Isola *et al.* [ILA15] studied the states and transformations of objects /scenes on image collections, and the learned state representations can be extended to different object classes. Fire and Zhu [FZ15] studied the causal relations between human actions and object fluent changes. Fathi and Rehg [FR13] developed a weakly supervised method to recognize actions and states of manipulated objects before and after the action. Wang *et al.* [WFG16] designed a Siamese network to model precondition states, effect states and their associate actions. Alayrac *et al.* [ASL17] optimized a discriminative cost for joint object state recognition and action localization.

5.3 Task-Fluent Dataset

Since there are no available public datasets for the proposed problem, we collected a new egocentric video dataset about tasks and object fluents. 14 volunteers performed daily tasks in 5 different indoor scenes freely with their own styles. A glasses camera, which can record the videos from the first-person view, is worn by the volunteers when they were performing the tasks. The video frame is at the resolution of 1280 × 960. Some frame samples in the dataset are shown in Figure 5.1.

In summary, our dataset consists of 809 videos with approximately 333,000 egocentric video frames. It contains 14 categories of tasks: *sweep floor, mop floor, write on blackboard, clean blackboard, use elevator, pour liquid from jug, make coffee, read book, throw paper, microwave food, use computer, search drawer, move bottle to dispenser*, and *open door*. These tasks involve 25 classes of objects: *broom, dustpan, trash, floor, bucket, mop, chalk, chalk box, blackboard, eraser, elevator, mug, jug, lid, coffee can, book, paper, trash can, microwave, food, monitor, drawer, bottle, dispenser*, and *door*.

In addition to tasks and related objects, this dataset contains 21 categories of object fluents, as shown in Table 5.1. These fluents are divided into self fluents and mutual fluents. In this dataset, though some objects have the same name of fluent values, they are regarded as different fluent values since their related objects are different. For example, *clean* with respect to *floor* and *clean* with respect to *blackboard* are regarded as different fluent values.

We manually annotated the task label for each video sequence and the object fluent labels in each video frame. A single video frame may contain multiple object fluents and all of the task related fluents are annotated.

Table 5.1: Object fluent categories.

Object	**Fluent**
Single Object: Self Fluents	
floor	clean / dirty
blackboard	clean / dirty
elevator	open / closed
microwave	open / closed
door	open / closed
book	open / closed
drawer	open / closed
mug	empty / filled / full
paper	complete / split
monitor	on / off
Two Objects: Mutual Fluents	
broom, trash	contacting / apart
eraser, blackboard	contacting / apart
bottle, dispenser	contacting / apart
bucket, mop	containing / separate
dustpan, trash	containing / separate
chalk box, chalk	containing / separate
trash can, paper	containing / separate
microwave, food	containing / separate
lid, coffee can	fastened / unfastened
bottle, dispenser	aligned / misaligned
mug, jug	coordinated / uncoordinated

5.4 Hierarchical Model of Tasks and Fluents

We use a hierarchical graph to represent tasks and object fluents in a unified framework, as shown in Figure 5.2. In this representation, a task is composed of several concurrent object fluent changes over time. These object fluents closely interact with each other both in spatial and temporal domains. For example, in Figure 5.2, the task *sweep floor* is composed of three categories of fluents. As time flows, the *floor* changes from *dirty* to *clean*; the group of *dustpan* and *trash* changes from *separate* to *containing*, and the group of *broom* and *trash* changes between *apart* and *contacting.* These fluents define the task from the perspective of the effects caused by human activities.

5.4.1 Definition

Task. $\mathcal{Y}$ is an alphabet containing K task category labels, such as *sweep floor*, *mop floor*, etc. There are a total of 14 task categories in our work, i.e. $K = 14$. Task recognition is to assign an optimal task label for an input video sequence from the K values in $\mathcal{Y}$.

Fluent. Let $\mathcal{F} = \{F_m|m = 1, \ldots, M\}$ be the set of all fluent categories, where M is the number of fluent category and 21 in our work, as shown in Table 5.1. F_m is an alphabet which denotes a fluent category, such as *dirty* or *clean* with respect to *floor.* The elements in F_m are the possible fluent values, such as '*dirty*' and '*clean*'.

5.4.2 Formulation

Let $X = \{\mathbf{x}_t|t = 1, \ldots, T\}$ be a video sequence containing T frames, where $\mathbf{x}_t$ is the video frame at time t. Suppose $Y \in \mathcal{Y}$ is the task category label of the sequence X. $Z = \{\mathbf{z}_t|t = 1, \ldots, T\}$ is the sequence of the fluent labels for the video sequence X.

For a specific task class $Y \in \mathcal{Y}$, it has N_Y categories of key fluents $\{F_{Y,n}|n = 1, \ldots, N_Y\}$, where $F_{Y,n} \in \mathcal{F}$ is the nth key fluent category of the task Y. For example, the task *sweep floor* has three categories of key fluents, as the three colorful bars show in Figure 5.2.

$\mathbf{z}_t = \{z_t^n|n = 1, \ldots, N_Y\}$ is the fluent label set of the video frame $\mathbf{x}_t$. The N_Y elements

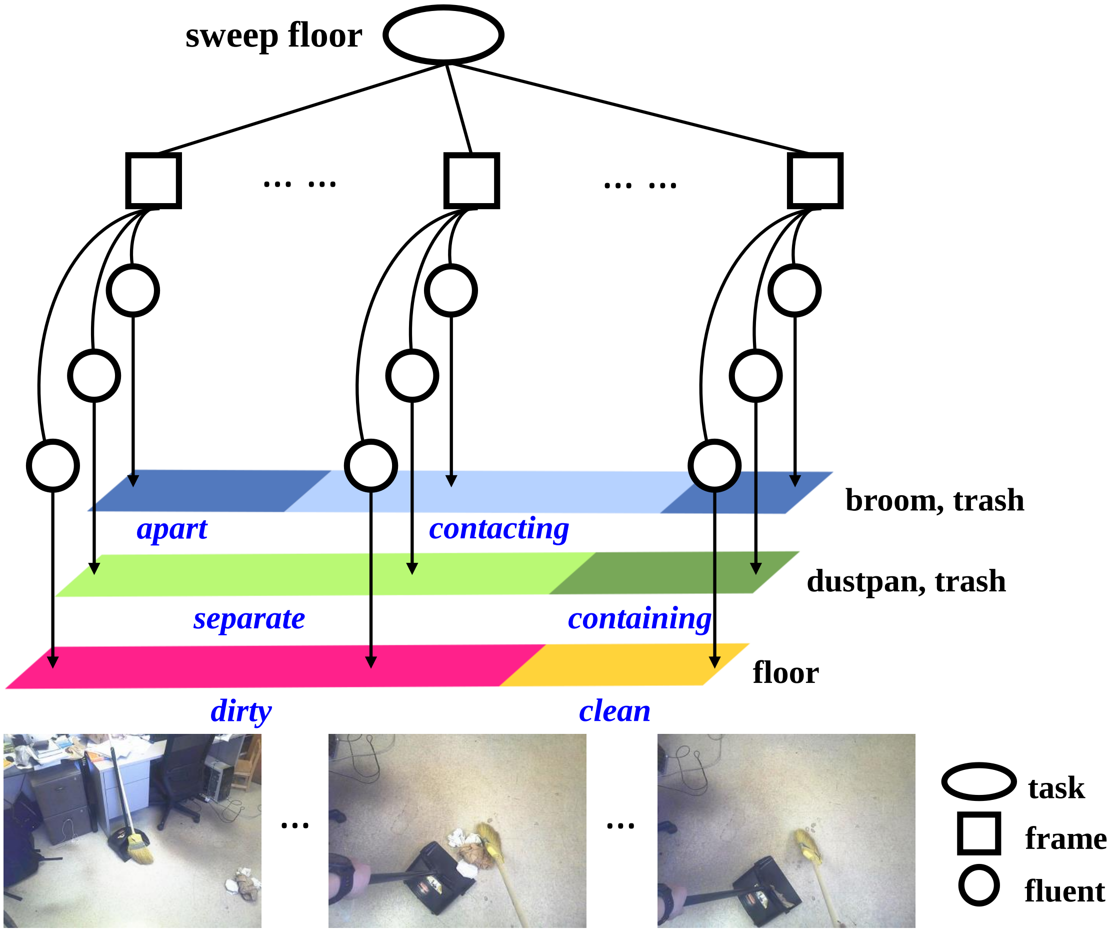


Figure 5.2: Joint model of tasks and object fluents.

of $\mathbf{z}_t$ correspond to the N_Y key fluents, respectively. $z_t^n \in F_{Y,n}$ is the value of the nth key fluent, such as $z_t^n =$ '*clean*' in the fluent with respect to *floor*, $z_t^n =$ '*apart*' in the fluent with respect to *broom* and *trash*, etc.

The score that the video sequence X is interpreted by the task category label Y and the fluent label sequence Z is defined as

$$S(X,Y,Z) = \underbrace{\sum_{t=1}^{T}\sum_{n=1}^{N_Y}\varphi(\mathbf{x}_t, z_t^n, Y)}_{\text{feature matching}} + \underbrace{\sum_{t=1}^{T}\sum_{n\neq n'}^{N_Y}\phi(Y, z_t^n, z_t^{n'})}_{\text{spatial concurrence}} + \underbrace{\sum_{t=2}^{T}\sum_{n=1}^{N_Y}\psi(Y, z_{t-1}^n, z_t^n)}_{\text{temporal transition}}, \quad (5.1)$$

where $\varphi(\cdot)$, $\phi(\cdot)$, and $\psi(\cdot)$ are the feature matching, spatial concurrence, and temporal transition functions, respectively. We elaborate on them as follows.

Feature matching. $\varphi(\mathbf{x}_t, z_t^n, Y)$ measures the compatibility between the fluent label z_t^n and the frame feature $\mathbf{x}_t$. Suppose $\mathbf{o}_{Y,n} = \{o_{Y,n}^i | i = 1, ..., |\mathbf{o}_{Y,n}|\}$ is the class label set of the related objects to the nth key fluent in the task Y, where $|\mathbf{o}_{Y,n}|$ is the related object class number. $|\mathbf{o}_{Y,n}|$ is 1 or 2 in our dataset. $\hat{\mathbf{o}}_{Y,n}$ is a set of bounding boxes of the objects in $\mathbf{o}_{Y,n}$.

$\varphi(\mathbf{x}_t, z_t^n, Y)$ is rewritten as

$$\varphi(\mathbf{x}_t, z_t^n, Y) = \varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n}) + \varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n}). \tag{5.2}$$

$\varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n})$ is the object detection term, which describes the occurrence belief of the fluent-related objects in the video frame at time t. We trained object detectors by fine-tuning Faster R-CNN [RHG15] on our dataset and use the trained detectors to generate the object detection probabilities. Suppose $p(o_{Y,n}^i | \mathbf{x}_t)$ is the detection probability of the object class $o_{Y,n}^i$. The object detection term is

$$\varphi_1(\mathbf{x}_t, \mathbf{o}_{Y,n}) = \frac{1}{|\mathbf{o}_{Y,n}|} \sum_{i=1}^{|\mathbf{o}_{Y,n}|} \ln p(o_{Y,n}^i | \mathbf{x}_t). \tag{5.3}$$

$\varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n})$ is the fluent labeling term, which measures the compatibility between the object state feature and the fluent label. We define the fluent area as a bounding box which covers all the bounding boxes in $\hat{\mathbf{o}}_{Y,n}$ with the minimum size. Using the features in the fluent areas, we train a classifier for each fluent category with VGG-16 model [SZ14b]. Suppose $p(z_t^n | \mathbf{x}_t, \hat{\mathbf{o}}_{Y,n})$ is the classification probability output by the fluent classifier. The fluent labeling term is defined as

$$\varphi_2(\mathbf{x}_t, z_t^n, \hat{\mathbf{o}}_{Y,n}) = \ln p(z_t^n | \mathbf{x}_t, \hat{\mathbf{o}}_{Y,n}). \tag{5.4}$$

Spatial concurrence. $\phi(Y, z_t^n, z_t^{n'})$ measures the compatibility between different fluent categories z_t^n and $z_t^{n'}$ in task Y. For each task category, we compute the average prior frequencies that the values of different fluents occur together from the training videos. Suppose $q(z_t^n, z_t^{n'})$ is the average prior frequency that z_t^n and $z_t^{n'}$ occur together. The spatial concurrence term is defined as:

$$\phi(Y, z_t^n, z_t^{n'}) = \ln q(z_t^n, z_t^{n'}) \tag{5.5}$$

Temporal transition. $\psi(Y, z_{t-1}^n, z_t^n)$ measures the continuity and transition relations of the fluent values z_{t-1}^n and z_t^n in two adjacent frames. We use a Markov chain to model the fluent value transitions. Suppose $r(z_{t-1}^n, z_t^n)$ is the probability of the transition from z_{t-1}^n to z_t^n. The temporal transition term is

$$\phi(Y, z_{t-1}^n, z_t^n) = \ln r(z_{t-1}^n, z_t^n). \tag{5.6}$$

The transition probabilities are learned from video samples of each task.

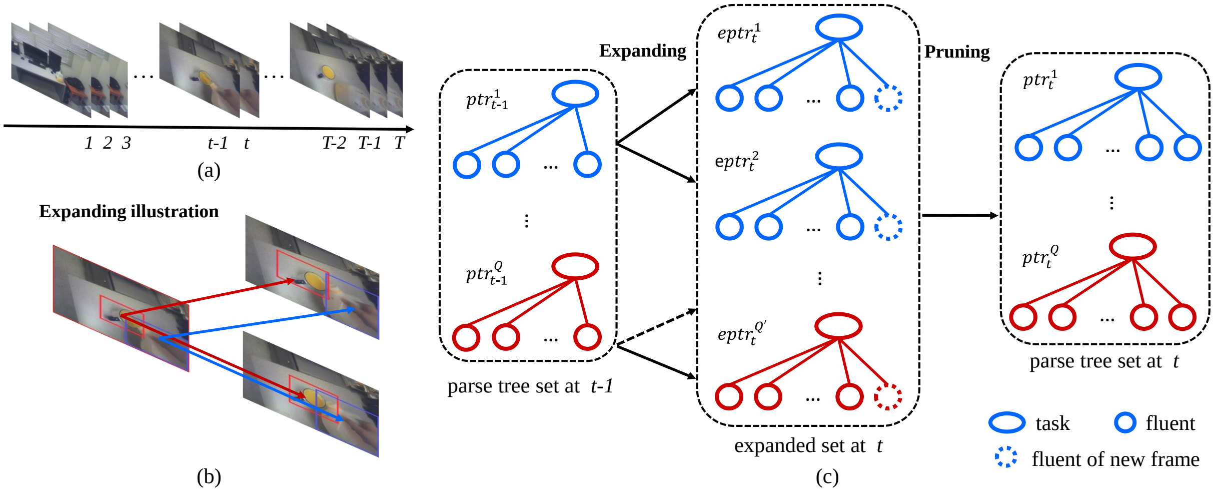


Figure 5.3: Dynamic programming beam search for jointly recognizing fluents and tasks. (a)The input video sequence. (b) Illustration of expanding one parse tree to new parse trees. (c) Parse tree expanding and pruning in one iteration.

5.5 Inference

Given a video sequence $X = \{\mathbf{x}_t | t = 1, \ldots, T\}$, the goal is to compute its task label Y^* and the fluent label sequence $Z^* = \{\mathbf{z}_t^* | t = 1, \ldots, T\}$ that maximize the score function

$S(X, Y, Z)$, which is formulated as

$$(Y^*, Z^*) = \underset{Y,Z}{\arg\max}\ S(X, Y, Z) \tag{5.7}$$

The solution to Eq. 5.7 is a parse tree, in which the root node is the task label, and the leaf nodes are the sequence of fluent labels. To obtain the optimal result, an intuitive idea is to examine all the possible parse trees and output the one with the maximum score as the optimal result. However, the huge size of the parse tree space makes such exhaustive search method inapplicable. Inspired by the work in [WZZ17], we adopt a beam search algorithm to solve the above problem 5.7.

The general procedures of this algorithm include: (1) proposing multiple object bounding boxes with pre-trained object detectors in each video frame; (2) generating possible interpretations of the current frame by expanding the parse trees of previous frames; (3) pruning the parse trees with smaller scores and keeping the rest as the possible interpretations to the current video sequences. The algorithm is illustrated in Figure 5.3.

At the first frame, we enumerate all possible task labels and the corresponding fluent labels based on the task-related object proposals to initialize a parse tree set. At time $t-1$, suppose $PTR_{t-1} = \{ptr_{t-1}^i | i = 1, \ldots, Q\}$ is the parse tree set with Q parse trees of labeling the video clip from time 1 to time $t-1$. With the frame at time t, we expand every parse tree $ptr_{t-1}^i \in PTR_{t-1}$ by adding new task-related object proposals and key fluent values in the new frame, as shown in Figure 5.3 (b). After expanding all $ptr_{t-1}^i \in PTR_{t-1}$, we obtain an expanded parse tree set $\{eptr_t^1, ..., eptr_t^{Q'}\}$ for the video clip from time 1 to time t, as shown in Figure 5.3 (c).

The expanded set $\{eptr_t^1, ..., eptr_t^{Q'}\}$ often contains large number of parse trees and many of them with small scores are misleading interpretations to the video clip. To increase the computational efficiency and improve the performance, we sort all the parse trees in the expanded set by their scores and keep the first Q trees with the largest scores. In this way, we obtain the parse tree set $PTR_t = \{ptr_t^i | i = 1, \ldots, Q\}$ at time t, as is shown in Figure 5.3 (c). This expanding and pruning process are iterated to the last frame of the sequence. The optimal (Y^*, Z^*) for the entire sequence is the parse tree in PTR_T with the maximal score.

5.6 Experiments

5.6.1 Experiments setup

We test our method on our newly collected Task-Fluent Dataset and the evaluations include fluent recognition and task recognition. We use recognition accuracy as the evaluation metric. For fluent recognition, the accuracy is defined as the ratio of the correctly recognized fluent state number to the total testing fluent state number in all testing video frames. For the task recognition, the accuracy is defined as the ratio of correctly recognized video number to the total testing video number. The ratios for the training, validation, and testing video numbers are 0.5, 0.25, and 0.25, respectively.

For object detection and proposal in Eq. 5.2, we fine-tune Faster R-CNN [RHG15] on our dataset using VGG-16 [SZ14b] trained on ILSVRC2012 [RDS14] as pre-trained model. The number of iterations we set for 2 stage training process are 80K, 40K, 80K, 40K. The confidence threshold is 0.5 and the non-maximum suppression threshold is 0.6. Some object proposal results are shown in Figure 5.4.

We use the ground truth object bounding boxes and fluent labels from the dataset to train the fluent feature matching term in Eq. 5.2. For each fluent category, we crop the fluent areas from video frames and trained classifiers by fine-tuning the VGG-16 model [SZ14b].

5.6.2 Fluent recognition

We compare our method with several baseline methods. (1) Single frame classification (SFC). This method takes fluent recognition as a multi-class image classification problem. We train an classifier by fine-tuning VGG-16 model [SZ14b] and replace *soft-max* with *sigmoid* to adapt for multi-class output. We select the threshold = 0.1 by maximizing the fluent classification accuracy on validation set. (2) Spatial stream LSTM (Spatial LSTM). This method uses a LSTM network [Gra12] with image appearance features for fluent recognition. Based on the network we train in method SFC, for each frame in the video, we retrieve 4096 dimensional features from *fc2* as the appearance descriptors. The videos are trimmed into

Table 5.2: Comparison of different methods for fluent recognition.

Method	**Self**	**Mu-tual**	**Over-all**
SFC	0.838	0.836	0.837
Spatial LSTM	0.859	0.820	0.843
Two-s LSTM	0.861	0.846	0.855
Our Method	**0.909**	**0.869**	**0.896**

video clips with length L = 60. The LSTM network was built by stacking three bidirectional LSTM layers. (3) Two-stream LSTM (Two-s LSTM). Inspired by Two-stream networks [SZ14a] and two recently studies which applied multi-stream to egocentric video action recognition [MFK16, SAJ16], we use the two-stream LSTM for fluent recognition. We train a multi-class classification network on optical flow features with a network architecture similar to VGG-CNN-M [CSV14] as the temporal stream LSTM. We build a two-stream LSTM network by combining the temporal LSTM and the spatial stream LSTM.

Table 5.3 and Table 5.4 show the accuracy comparison on each fluent category. Our method achieves best performance on most fluent categories, which demonstrates the advantage of our model.

Table 5.2 shows the overall accuracy comparison of our method with other baseline methods. It also separately shows the self fluent recognition accuracy and mutual fluent accuracy. Our method outperforms other baseline methods in each item, which demonstrates the advantage and effectiveness of our joint modeling method. This table also shows that, to each method, the recognition accuracy on self fluents is higher than that of mutual fluents. One explanation is that the mutual fluents are related to the spatial relationships between two objects. It is difficult to encode such spatial relationships on 2D images.

Some qualitative results are shown in Figure 5.4. In most of the cases, our method can identify the task category, locate the objects, and recognize the fluents correctly. It should be noted that our method can infer the object fluents without the fluent features. For example,

Table 5.3: Comparison of different methods for self fluents.

Method	**floor**	**board**	**elevator**	**mcwave**	**door**	**book**	**drawer**	**mug**	**paper**	**monitor**
	clean	*clean*	*open*	*open*	*open*	*open*	*open*	*empty*	*complete*	*on*
	dirty	*dirty*	*closed*	*closed*	*closed*	*closed*	*closed*	*filled/full*	*split*	*off*
SFC	0.79	0.66	0.98	0.79	0.78	0.93	0.75	0.85	0.91	0.97
Spatial LSTM	0.85	0.67	0.99	0.90	0.83	0.90	0.82	0.87	0.86	**0.99**
Two-s LSTM	0.84	0.70	0.99	0.91	**0.94**	0.88	0.67	**0.89**	0.90	**0.99**
Our Method	**0.90**	**0.76**	**1.00**	**0.92**	0.86	**0.99**	**0.99**	0.66	**0.94**	0.98

Table 5.4: Comparison of different methods for mutual fluents.

Method	**broom**	**eraser**	**bucket**	**dustpan**	**box**	**mug**	**trashcan**	**mcwave**	**lid**	**bottle**	**bottle**
	trash	**board**	**mop**	**trash**	**chalk**	**jug**	**paper**	**food**	**coffcan**	**dispens**	**dispens**
	contact	*contact*	*contain*	*contain*	*contain*	*coord*	*contain*	*contain*	*fastened*	*contact*	*aligned*
	apart	*apart*	*sep*	*sep*	*sep*	*uncoord*	*sep*	*sep*	*unfstn*	*apart*	*misalign*
SFC	0.58	0.89	0.87	0.67	**0.84**	0.86	0.93	0.89	0.82	0.80	0.87
Spatial LSTM	0.51	0.92	0.77	0.86	0.74	0.80	0.94	**0.94**	0.78	0.82	**0.92**
Two-s LSTM	0.73	0.89	0.86	0.81	0.78	0.71	**0.99**	0.89	**0.87**	0.83	0.88
Our Method	**0.88**	**0.95**	**0.98**	**0.90**	0.81	**0.95**	0.93	**0.94**	0.58	**0.89**	**0.92**

* board = blackboard; box = chalk box; mcwave = microwave; coffcan = coffee can; coord = coordinate; sep = separate; unfstn = unfastened.

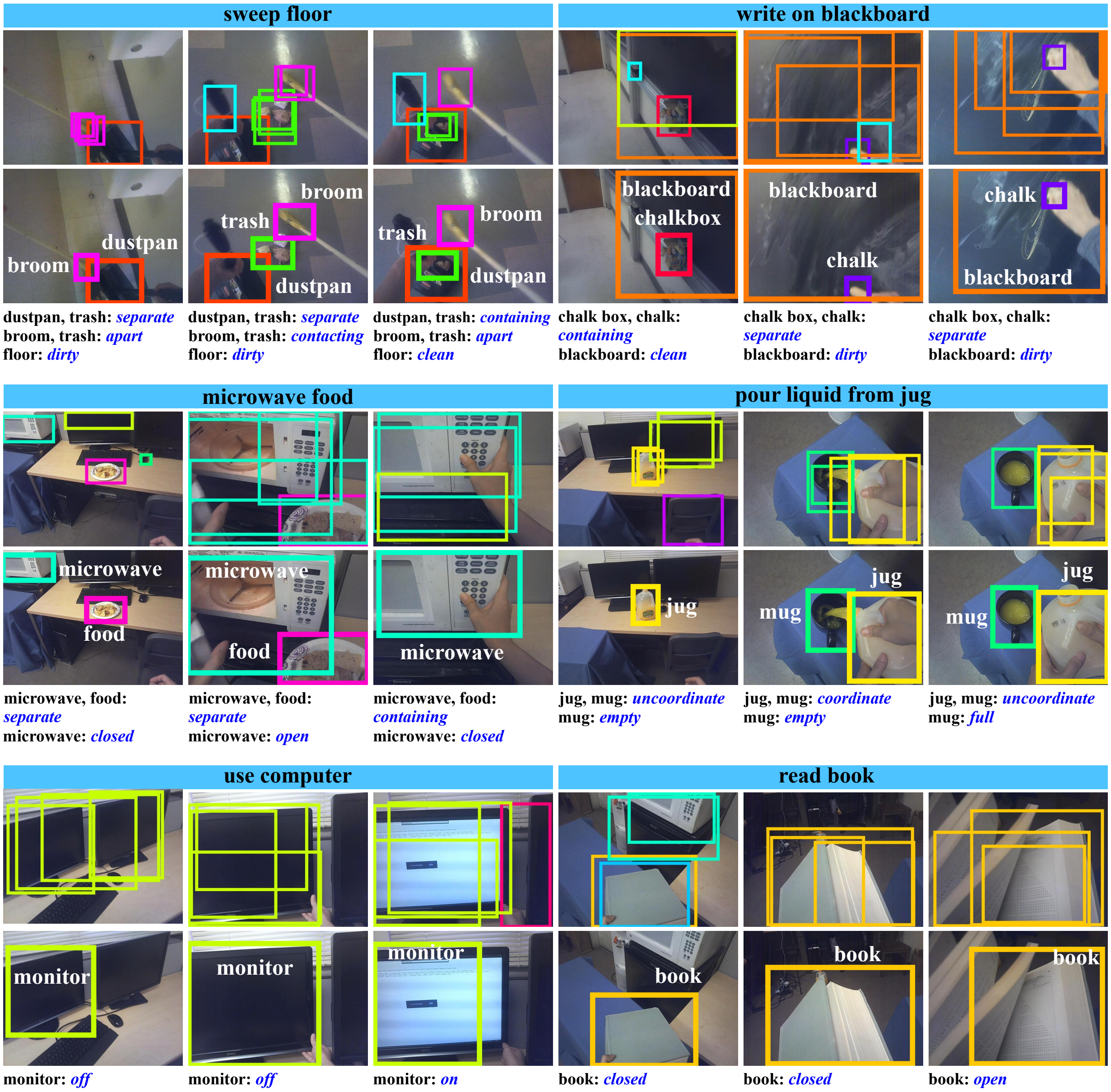


Figure 5.4: Some results of joint recognition of object fluents and tasks. The labels in the bars are the tasks. For every two rows, the first row shows the object proposals; the second row shows our output results, and the descriptions below the images are about the fluent recognition.

Table 5.5: Comparison of different methods for tasks recognition

Method	**Average**
CNN hit@1	0.90
LSTM	0.87
Our Method	**0.96**

in the first image of the task *sweep floor*, although *trash* is not in the frame and *floor* is not detected, our method can still infer the correct fluent label by reasoning about the spatial concurrence and temporal transition relations among fluents. In the task *read book*, task-unrelated object *microwave* is detected in the frame. Our joint model correctly recognizes the task and outputs correct fluent label of *book*. This illustrates the advantage of the joint modeling framework of tasks and fluents.

5.6.3 Task recognition

For task recognition, we compare our method with two baseline methods. (1) CNN hit@1. This method uses the single frame features with hit@1 rule [KTS14] to recognize tasks of videos. As described in Karpathy's work on video classification [KTS14], we train a single frame classifier by fine-tuning VGG-16 model with a 14-dimensional output layer at top. In testing, each frame can output one task label, we average those predictions of frames and output one explicit task label for each sequence. (2) LSTM. As the approach suggested in [YHV15], we retrieve the output of the second fully connected layer of single frame model we train in CNN hit@1. The video-level three stacked LSTM networks are trained with video clips of length 10 for task recognition.

As shown in Table 5.5, our method outperforms the baseline methods, which shows that the fluent-based representation of complex tasks is reasonable and effective. Table 5.5 shows that the CNN hit@1 method achieves a better accuracy than the LSTM method incorporating the motion information. The main reason is that the egocentric videos contain massive motion features which are not related to tasks but caused by the irregular movement of the user's head. Such motion information will mislead task recognition.

Since our model has a hierarchical structure and our inference algorithm is an online framework, our method can recognize the task with partial observation of a video sequence, i.e. early recognition of tasks. We compare our method with above 2 baseline methods at different observation ratios of each video sequence. At each ratio point, each method is fed with a video clip from the first frame to the frame at the position corresponding to the ratio length of the whole video length.

Figure 5.5 shows the accuracy comparison at different sequence length ratios. Our method outperforms the other baseline methods at every observation ratio. This figure shows that with fewer observation video frames our method can achieve a comparable accuracy with other methods at a lager observation ratio. This is mainly because our fluent-based method uses features of objects related to the tasks rather than the entire image features.

5.7 Conclusion

In this work, we study a new problem of jointly recognizing object fluents and tasks in egocentric videos. We propose a unified fluent-based task representation framework, in which tasks are modeled with object fluents. In each task, different fluents closely interact with each other by means of spatial concurrence and temporal transition. Given a testing egocentric video, a beam search algorithm is used to jointly recognize the object fluents in each frame, and the task of the entire video. We collect a large-scale egocentric video dataset including various fluents and tasks with detailed annotations. Our experiments have shown that our model outperforms the baseline methods which proves the strength of our model. Our future work will focus on the continuous fluents, object independent fluents, and the fluent-based tasks in robotics.

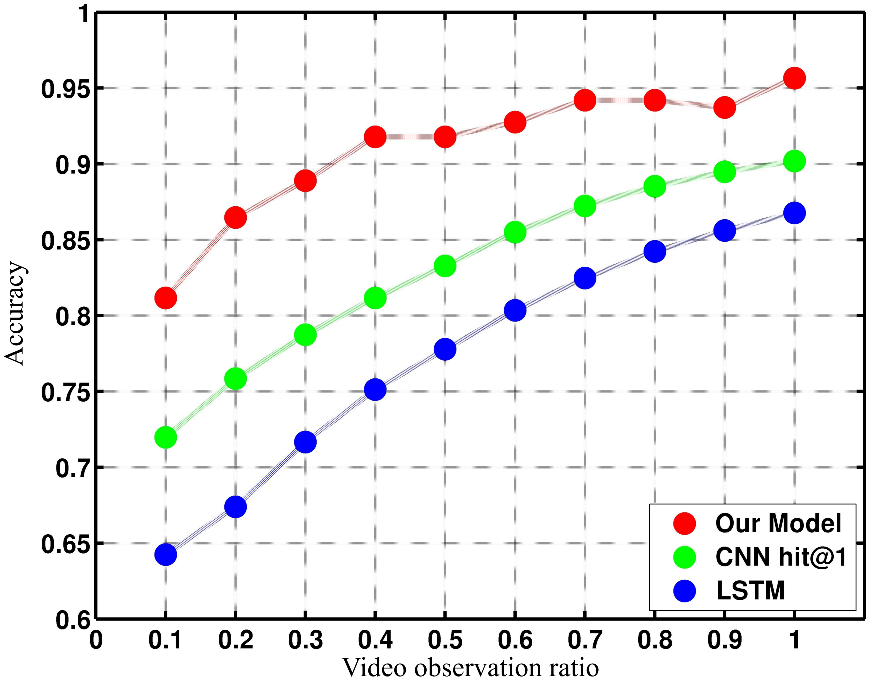


Figure 5.5: Comparison of our method with baselines.

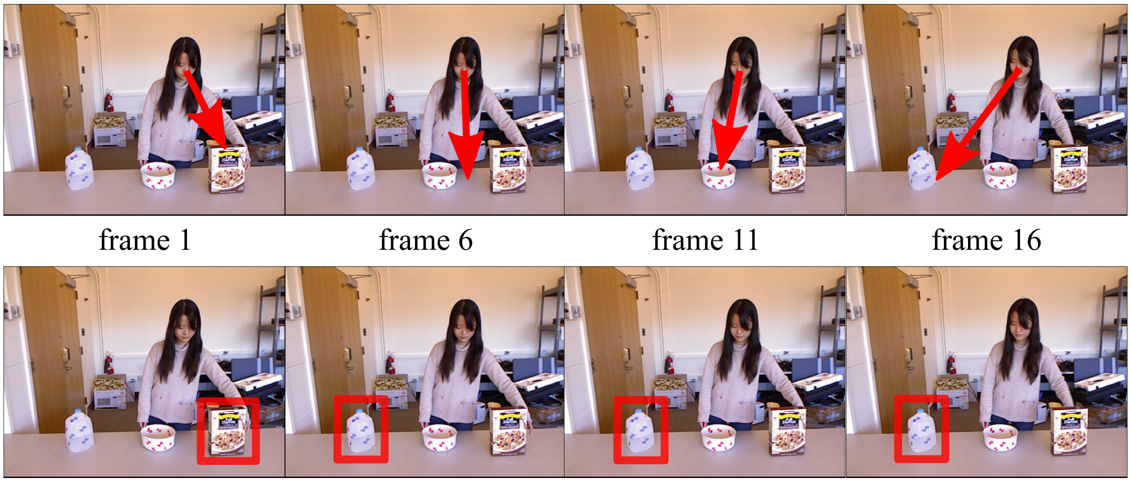
CHAPTER 6

Task Representation with Human Attention

6.1 Introduction

The human gaze, indicating where a human is physically looking at, has the clear definition and is usually defined as a direction [PSH18, WBM16, ZSF15, ZSF17b], a location [RKV15], or a direction along with a location [WXZ17]. However, attention does not have an universally accepted definition. In the past 30 years, visual attention mainly refers to the eye fixation saliency map or saliency object, which is actually the attention of a human outside images who is looking at images [IKN98, IK00, IK01, Itt05, BAA10, BSI14, ZZW16, WS18]. However, there is another case that humans inside images have their own attention. Therefore, human attention can be roughly divided into two categories: 1) the attention of a human outside images, and 2) the attention of a human inside images. In this work, we are studying the attention of a human inside images, which is usually confused with human gazes.

The foremost thing to clarify is what human attention is and how it differs from human gazes. Originally, attention is a concept in philosophy. Nowadays, it is well known as a concept in psychology. One dominant definition in psychology is that attention is something that happens in the mind - a mental “inside” which is linked with the perceivable “outside” [See11]. This definition indicates that attention is an internal invisible mental state in the human mind, but it is linked with visible things. Another widely accepted definition is that attention is the process of attending to objects [EDR94, Sch01]. This definition indicates that the attention is based on objects. Actually, some other studies in psychophysics and biology [Che12, CY12, PR14] as well as some inter-discipline studies such as neuro image [ZMJ17]



frame 1 frame 6 frame 11 frame 16

frame 1 frame 6 frame 11 frame 16

Figure 6.1: The arrows in the upper row images represent the human's gaze, and the bounding boxes in the lower row images represent the human's attention. The images are sampled from a video in the public CAD120 dataset [KGS13]. From 1^{th} frame to 6^{th} frame, we can observe that her left hand is leaving the cereal box. Based on the task the human is doing, her hands will then approach to the milk box. Therefore, her attention has changed from the cereal box to the milk bottle at 6^{th} frame. However, we can observe that the human is not gazing at her attentional object between 6^{th} and 16^{th} frame.

and brain image [MTV06, MK03, SB06] also claim the object-based attention. Especially, Chou *et al.*[CY12] provides the evidence of object-based attention. These studies provide the strong theory basis for object-based attention definition in computer vision.

Based on these studies, we define human attention as the attentional objects that a human is operating on or attending to. The attentional objects coincide with the human mind, and are driven by the task a human is doing. The most significant difference between the gaze and attention is that the gaze is what a person is physically gazing at while attention might be the objects that a person is attending to but not necessarily gazing at. For example, as shown in Figure 6.1, there are four images in each row, sampled from a video in which the human is doing the task of making cereal. Based on the task, after her left hand leaves the cereal box, her hand will approach to the milk bottle. Therefore, her attention (the bounding boxes in the lower row images) has shifted from the cereal box to the milk bottle when her left hand begins to leave the cereal box. We can observe that her attention is not always what she is gazing at (the arrows in the upper row images represents the gazes). Factually, when a human is doing a complex task in large scenes, the attention may even be the objects that a human is not operating on, or the objects that are far away from or at the behind of the human. That is because human attention is driven by the ongoing task, coinciding with the human mind which usually goes ahead of human gazes and human actions. As a result, in many cases, attention is the objects that a human will gaze at or operate on in the future. Benefiting from this characteristic, human attention is significant for many applications. For example, it enables a robot to estimate the attentional objects that a human will operate on in the short future, so that the robot can assist the human in advance or cooperate with the human to finish the task.

Our method is based on two observations. First, human pose and motion significantly signal human attention. For example, the objects, which a human is approaching to, are more likely to be the attentional objects. Second, human attention is driven by the task. Given a task, humans know what to do now and what to do next in the mind. This is also why, in some cases, a human's attention has shifted to another object even when he is still operating on or gazing at the current object. Based on these two observations, we propose a model that integrates low-level human pose and motion cues and high-level task information.

Our contributions are three-fold: 1) Different from previous studies which equate human gazes as human attention, we formulate attention as the attentional objects that coincide with the human mind. 2) We propose an architecture that incorporates both low-level human body cues and high-level task information into a unified framework, and the code will be released. 3) We collected and annotated two large-scale video datasets, and also re-annotated a public dataset. To the best of our knowledge, the collected datasets are both new ones for human attention research. The datasets will also be publicly available.

6.2 Related work

In this section, we first review several related problems and the typical methods for solving these problems. Then we analyze the datasets that are widely used for studying these problems.

6.2.1 Related problems and methods

Gaze has two basic categories, first-person view gaze [FLR12, LWZ17] and third-person view gaze [PBI15, RKV15, WXZ17, WLS18]. Classic gaze estimation methods usually operate in the bottom-up manner. Low-level visual features extracted from human pupil, eye and face are fed to a model to regress a gaze direction [PSH18, PZB18, ZSF15, ZSF17a] or used to fit to a known model [WJ17, WBM16] to find the most possible gaze. In this work, we estimate the attention of a human in the third-person view.

Saliency estimates the saliency object or eye fixation saliency map that signals the regions of an image where human observer would pay attention at first glance. The typical pipeline is to first predict a saliency map [KTB14, HSB15, KAB17, WS18], and then minimize the difference between the saliency map prediction and the ground truth. The ground truth of saliency is obtained based on human eye fixation locations recorded by the humans wearing the eye-tracking equipment [WS18]. Therefore, saliency signals the attention of a human outside images. Different from these studies, in this work, we study the attention of a human inside images.

Human object interaction (HOI) involves two fundamental problems, HOI classification and HOI detection. Given an image, the former outputs a binary label for each HOI category, while the latter outputs a triplet of the human, object and HOI label [CLL18]. The typical HOI detection and classification methods are based on the bottom-up information that encodes the spatial relation between human skeleton key points and the object [KGS13], extracts the CNN features from the bounding box enclosing both the human and the object [QWJ18], or exploits the global context from the whole image [ML16]. Most HOI studies are limited in still images, and the involved objects are those a human is directly interacting

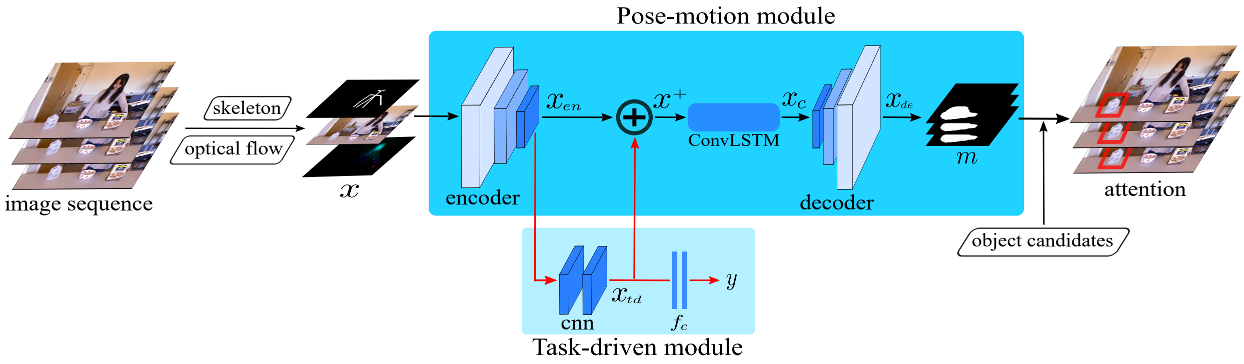


Figure 6.2: Overview of our method. Given an image sequence as input, the output is human attention (a set of attentional objects marked by the red bounding boxes on the output image sequence). Our model consists of two basic modules, the pose-motion module and the task-driven module. The pose-motion module is composed of three sub-modules, "encoder", "ConvLSTM" (Convolution Long Short Term Memory), and "decoder". The task-driven module is composed of two sub-modules, "cnn" (convolution neural network) and "fc" (fully connected neural network layers). The input image is concatenated with two human body cues (human skeleton and optical flow), generating the concatenated feature map x to serve as the input of "encoder". The output x_{en} of the "encoder" and the output x_{td} of the "cnn" are fused as x^+, which is then processed by "ConvLSTM" and "decoder" to output the attention heat map m (a probability matrix). Human attention is finally estimated based on the attention heat map and object candidates.

with at the current time. In this work, we study the attention of a human who is doing a task in a video, and attentional objects might be the distant objects that a human is not currently interacting with or gazing at.

Human action recognition/prediction estimates the action label from a video containing complete action execution or predicts the action label from an incomplete video[KF18]. Human action recognition has witnessed the significant progress from hand-designed feature based methods (*e.g.*the Improved Dense Trajectory (IDT) [WS13]) to deep learning based methods. One successful deep learning method, termed as two-stream, is proposed in 2014 [SZ14a], with one stream to extract spatial information and the other stream to extract temporal motion information. Beyond 2D models, 3D deep convolution neural network based models (*e.g.*I3D [CZ17]) have achieved better performance. Recently, graph-based reasoning has sparked the interest of researchers [WG18]. Different from human action recognition and prediction that infer human action labels, in this work, we estimate attentional objects.

6.2.2 Related datasets

Gaze datasets. EYE-DIAP [FMO14], MPIIGaze [ZSF17b], and Columbia Gaze [SYF13] are three benchmark datasets. These datasets are collected in simple scenarios and the humans inside images are restricted with limited head and body movements. As a result, the detail facial information (*e.g.*pupil, eye, and face) of a human is fully observed. To stride to large and complex scenes where humans are moving freely and the detail facial information is not always available, some challenging and natural datasets like GazeFollow [RKV15], Flickr gaze [PBI15] and VideoGaze[RVK17] are proposed. However, the annotations of these datasets are gaze locations rather than objects, and no task is involved.

Saliency datasets. MIT1003 [JED09], TORONTO [BT06], PASCAL-S [LHK14], and DUT-OMRON [YZL13] are four widely used datasets. These datasets are related with human attention. However, they are proposed for studying the attention of a human outside images.

HOI datasets. HICO-DET [CLL18] and V-COCO [GM15] are two benchmark datasets. They are not suitable for studying task-driven human attention for two reasons: 1) the datasets are composed of still images, and 2) the annotations are limited to the objects that a human is directly interacting with. However, in this work, we estimate the attention of a human in a video where a task is involved, and attentional objects might be the objects that a human is not directly interacting with at the current time.

Human action dataset. Human action is an important research topic, thus existing a large number of datasets such as KTH [SLC04], Hollywood2 [MLS09], and Kinetics [KCS17], to name a few. However, the annotations are action labels rather than objects.

As a conclusion, these datasets are not suitable for estimating the attention of a human inside images who is doing a task in a video. To our best knowledge, there exists no appropriate public dataset. Therefore, we collected and annotated two large-scale datasets. In addition, we re-annotated a public dataset, CAD120 [KGS13].

6.3 Approach

In this section, we start with the overview of our model, then introduce the outline of our network architecture, followed by the detail description of data transition flow, finally, we explain our loss function.

6.3.1 Overview

Given an image sequence as input, the output is the human attention in the image sequence. As shown in Figure 6.2, our model consists of two basic modules, the pose-motion module and the task-driven module. The two basic modules are composed of five sub-modules, 'encoder", "ConvLSTM", "decoder", "cnn" and "fc". The raw image, together with the cues of human skeleton and optical flow, are concatenated as x to serve as the input of the "encoder". The output x_{en} of the "encoder" is taken as the input of the "cnn" in the task-driven module. The output x_{td} of the "cnn" is fed back to the pose-motion module to fuse with the original output x_{en} of the "encoder", generating the fused feature map x^+. x^+ is processed by the "ConvLSTM" and the "decoder", outputting the attention heat map m. The attention heat map and object candidates are jointly inferred to finally output human attention.

6.3.2 Network architecture outline

Our network architecture, consisting of the pose-motion module and the task-driven module, is inspired by the studies that have demonstrated attention is controlled in both bottom-up and top-down manner [CS02, KU87]. The motivation of this architecture is two-fold. On one hand, the backward propagation of the task-driven module can update the parameters of the pose-motion module, guiding the network to predict the task-driven attentional objects.

On the other hand, the output of the task-driven module is fused with the low-level human body cues in the pose-motion module, allowing the network to predict attentional objects using both the low-level human body cues and the high-level task information.

The pose-motion module uses the encoder-decoder backbone, which is inspired by some classic works of semantic segmentation [BKC17, LSD15] and saliency estimation [WSS18]. In addition, we add a ConvLSTM (Convolution Long Short Term Memory) network [XCW15] between the encoder and the decoder. This "Encoder-Decoder + ConvLSTM" architecture is effective for spatial-temporal reasoning and some similar architectures have been widely used [OGL15, SMS15].

For the task-driven module, as shown in Figure 6.2, we design a two-layer convolution neural network ("cnn"), followed by two fully connected layers ("fc"). Studies like [CSV14] have indicated that deep layers tend to capture the global context information while shallow layers are more powerful to capture local detail information. "cnn" is two additionally deep convolution layers, so its output x_{td} is supposed to carry more high-level task information.

6.3.3 Data transition flow

Input is the image sequence $\mathcal{V} = \{I_t | t = 1, 2, ..., T\}$, and output is human attention $\mathcal{A} = \{a_t | t = 1, 2, ..., T\}$. For the convenience of expression, we omit the subscript t of all variables. Now, the input is the image I with the size of $3 \times H \times W$ (3 channels, H pixels in height, and W pixels in width), and the output is human attention a that is defined as attentional objects.

We first describe how we extract the low-level human body cues. Human skeleton is an effective cue that has been widely used in the studies of human-object interaction detection and human action recognition [WZZ17, SPS12, XLW18]. Inspired by these works, we use the method proposed in [CSW17] to extract human skeleton as one of our low-level human body cues, which is targeted to capture the human pose information. Another cue is the optical flow that captures human motion information. The motion significantly signals attentional objects. For example, when a human's hand is approaching to an object, this object is more

Sub-module	Input	Output	Input size	Output size	Layer
encoder	x	x_{en}	$6 \times H \times W$	$512 \times \frac{H}{32} \times \frac{W}{32}$	VGG16 [SZ14b]
cnn	x_{en}	x_{td}	$512 \times \frac{H}{32} \times \frac{W}{32}$	$512 \times \frac{H}{32} \times \frac{W}{32}$	Conv2D, $(512, 1 \times 1) \times 2$
ConvLSTM	$x^+ = [x_{en}, x_{td}]$	x_c	$1024 \times \frac{H}{32} \times \frac{W}{32}$	$1024 \times \frac{H}{32} \times \frac{W}{32}$	ConvLSTM cell, $(1024, 3 \times 3) \times 2$
decoder	x_c	x_{de}	$1024 \times \frac{H}{32} \times \frac{W}{32}$	$1 \times H \times W$	Deconvolution, $(N_1, N_2) \times 5$
fc	x_{td}	y	$512 \times \frac{H}{32} \times \frac{W}{32}$	$1 \times N_{task}$	Linear, (N_{td}, 4096)×1, (4096, N_{task})×1

Table 6.1: The data transition flow of five sub-modules in our network. In the items of 'Input' and 'Output', the notions of x, x_{en}, x_{td}, y, x^+, x_c and x_{de} are the same with that shown in Figure 6.2. In the items of 'Input size' and 'Output size', H and W are the height and width of the input image, and N_{task} is the task number. The item 'Layer' denotes the network layers of each sub-module. The sub-module "encoder" is the same with the VGG16 network [SZ14b]. The sub-module "cnn" is composed of 2 convolution layers, and each layer has 512 filters with the kernel size of 1×1. The sub-module "ConvLSTM" has 2 convolution layer in each ConvLSTM cell, each convolution layer has 1024 filters with the kernel size of 3×3. The sub-module "decoder" has 5 deconvolution layers, and each layer has different number of filters so that we use (N_1, N_2) for unified representation. The sub-module "fc" is composed of two fully connected layers, the first layer is with the size of (N_{td}, 4096) and the second layer is with the size of (4096, N_{task}), where N_{td} is the size of x_{td}.

likely to be the attentional object. In contrast, when a human's hand is leaving an object, this object is usually not the attentional object. Motion information is not included in the single still image. We use OpenCV to extract the optical flow using adjacent two images. As shown in Figure 6.3, the extracted human skeleton is binarized as a human skeleton mask h_s with the size of $1 \times H \times W$. Optical flow o_f is with the size of $2 \times H \times W$. We concatenate I, h_s, and o_f to obtain the fused feature map x with the size of $6 \times H \times W$:

$$x = [I, h_s, o_f] \tag{6.1}$$

$[\cdot, \cdot,]$ denotes the concatenation operation.

As shown in Figure 6.2, the data transition starts from x, then flows through five sub-modules: "encoder", "ConvLSTM", "decoder","cnn", and "fc". For a clearer representation, we summarize the data transition detail (including the input, output, data size, and network

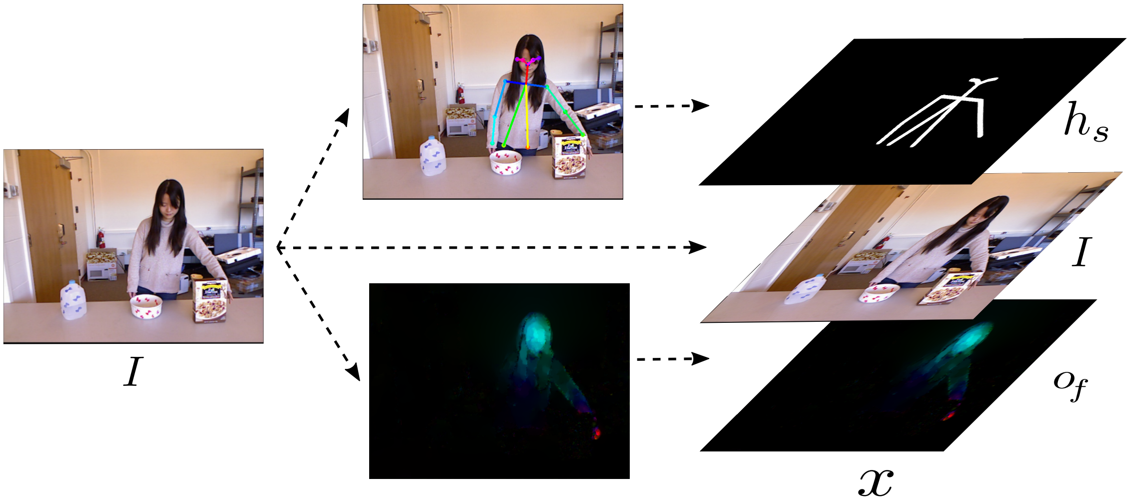


Figure 6.3: The visualization of the low-level human pose and motion cues. Given an image I, human skeleton is detected and binarized as a human skeleton mask h_s. The optical flow o_f is extracted using the input image I and its previously adjacent image. We concatenate I, h_s, and o_f together to generate the final feature map x.

layers) of the five sub-modules in Table. 6.1.

The output of the pose-motion module is the attention heat map m, which is computed by adding a sigmoid activation layer σ after x_{de}:

$$m = \sigma(x_{de}) \tag{6.2}$$

m is a probability map, and $m \in [0, 1]^{H \times W}$.

The output of the task-driven module is the task label y, which is computed by adding the fully connected layers $\mathcal{F}_l$ after x_{td}:

$$y = \mathcal{F}_l(x_{td}) \tag{6.3}$$

Human attention a is inferred based on the attention heat map m and the object candidates. The inference method will be detailed in the inference section.

6.3.4 Loss function

The loss $\mathcal{L}$ consists of the pose-motion loss $\mathcal{L}_{bu}$ and the task-driven loss $\mathcal{L}_{td}$:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{bu} + \lambda_2 \mathcal{L}_{td} \tag{6.4}$$

The pose-motion loss $\mathcal{L}_{bu}$ is computed based on the attention heat map m defined in Eq. 6.2 and the ground truth heat map M. $m \in [0,1]^{H \times W}$ is a probability map, and $M \in \{0,1\}^{H \times W}$ is a binary map with attention region R^+ assigned with '1' and non-attention region R^- assigned with '0'. Let m_{ij} be the $(i,j)^{th}$ entry of m, and M_{ij} be the $(i,j)^{th}$ entry of M. One classic loss function is computing the loss ($e.g.$binary cross entropy loss) for every local $(i,j)^{th}$ entry, and then add them up. However, in many cases, the proportion of the attention region R^+ and the non-attention region R^- is imbalanced, which will lead to the poor performance of this kind of loss function. Therefore, we design a loss function that considers the global distribution of the attention heat map.

The motivation of our loss function is encouraging m to cover R^+ as much as possible and R^- as less as possible. The probability summation of m inside and outside of R^+ can be respectively computed as:

$$P_{in} = \sum_{i,j} m_{ij} \cdot M_{ij} \ , \quad P_{out} = \sum_{i,j} m_{ij} \cdot (1 - M_{ij}) \tag{6.5}$$

To encourage the majority of heat map m to be inside of attention region R^+ rather than non-attention region R^-, the pose-motion loss $\mathcal{L}_{bu}$ is defined as:

$$\mathcal{L}_{bu} = \frac{P_{out}}{A_{R^-}} - \frac{P_{in}}{A_{R^+}} \tag{6.6}$$

A_{R^+} is the acreage of R^+ and A_{R^-} is the acreage of R^-.

The task-driven loss $\mathcal{L}_{td}$ is defined as standard cross-entropy loss:

$$\mathcal{L}_{td} = CE(y, Y) \tag{6.7}$$

y is the task label prediction defined in Eq. 6.3 and Y is the ground truth of task label.

Dataset	Video	Image	Train	Test	Attention	Object	Task	Scene	Human
TaskAttention	770	308,958	230,116	78,842	378,509	649,315	14	5	14
TaskAttention-VR	76	79,976	59,941	20,035	88,458	1,191,349	5	4	4

Table 6.2: The statistics of our datasets. Video: video number, Image: image number, Train: training image number, Test: testing image number, Attention: attention annotation number, Object: object annotation number.

6.4 Learning and inference

Let W be all parameters involved in the network. Learning the optimal parameter W^* is equal to minimize the loss defined in Eq. 6.4:

$$W^* = \arg\min_{W}(\lambda_1 \mathcal{L}_{bu} + \lambda_2 \mathcal{L}_{td}) \tag{6.8}$$

Our network is implemented with PyTorch. We use ADAM algorithm to learn the parameters, the initial learning rate is set as 0.001 and with 10% decay after 5 epochs. Each image sequence has 10 images, and the batch size is set as 2 with two NVIDIA TITAN X GPUs. λ_1 and λ_2 are empirically set as 2 and 1, respectively.

For inference, given the input image, the goal is to estimate the score of each object candidate being the attentional object. Let $\mathcal{O} = \{o_k | k = 1, 2, ..., N\}$ be N object candidates, detected by the RetinaNet [LGG17] pre-trained on the ImageNet dataset [DDS09] and fine-tuned on our datasets. The score S_{o_k} of k^{th} object candidate o_k is computed as:

$$S_{o_k} = \frac{\sum\limits_{(i,j)\in o_k} m_{ij}}{A_{o_k}} \tag{6.9}$$

A_{o_k} is the acreage of o_k. This factually computing the proportion of the probability summation of m inside of o_k.

6.5 Datasets

To our best knowledge, there exists no dataset that is targeted for studying the attention of a human inside images who is doing a task. Therefore, we collected two large-scale video datasets. One is called “TaskAttention”, which is collected in various real indoor and outdoor scenes like offices, classrooms and corridors. During collection, volunteers behave freely to finish various tasks in different scenes with diverse camera views. Another is called “TaskAttention-VR”, which is collected in the Virtual Reality (VR) scenes. We use Unreal Engine 4 (UE4) to build four different kitchen scenes that consist of a large number of objects. A human can perform different tasks using the objects. For example, to make orange juice, a human first takes an orange from the refrigerator, then uses a knife to cut the orange into pieces and puts them into a juicer. To our knowledge, this is the first VR dataset for studying human attention. Some statistics of our datasets are summarized in Table. 6.2 and some samples are shown in Figure 6.4.

We annotate attentional objects and other objects for each frame and the task label for each video. When annotating attentional objects, the foremost thing is to guarantee that attentional objects coincide with the human mind, which is driven by the task a human is doing. To this end, when a human has finished a sub-task, attentional objects are immediately annotated as the objects involved in the next sub-task, even if the human is not gazing at or operating on the attentional objects in the current image. As a result, many annotations are the objects far away from or at the behind of a human. This makes human attention estimation to be a challenging yet meaningful problem. Note that we annotate one or two attentional objects in each image. We split training set and testing set based on the humans. For each task, we guarantee the humans in training set and testing set are not overlapping.

In addition, we annotate a public dataset, CAD120 [KGS13], in which objects and task labels have been annotated so that we only need to annotate the attentional objects.

Methods	T1	T2	T3	T4	T5	T6	T7	T8	T9	All
GazeFollow [RKV15]	**0.45**	0.40	0.63	0.70	0.23	0.76	0.31	0.45	0.63	0.54
PRNet [FWS18]	0.30	0.74	**0.82**	0.59	0.41	0.71	0.36	0.80	**0.80**	0.66
Hopenet [RCR18]	0.19	0.80	0.67	0.73	0.31	0.51	0.48	0.55	0.57	0.59
Our	0.36	**0.96**	0.74	**0.77**	**0.68**	**0.68**	**0.89**	**0.85**	0.79	**0.78**

Table 6.3: Accuracies on the CAD120 dataset. "All" corresponds to the overall accuracy. T1 to T9 correspond to the task categories. T1: arranging objects, T2: cleaning objects, T3: making cereal, T4: microwaving food, T5: picking objects, T6: stacking objects, T7: taking food, T8: taking medicine, and T9: unstacking objects.

Methods	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	All
GazeFollow [RKV15]	0.46	0.27	0.45	**0.68**	0.80	0.32	0.37	0.64	0.57	0.79	0.11	0.52	0.63	0.24	0.51
PRNet [FWS18]	0.24	0.36	0.33	0.38	**0.83**	0.17	0.41	0.77	0.35	0.59	0.53	0.79	0.56	0.58	0.52
Hopenet [RCR18]	0.16	0.22	0.10	0.24	0.73	0.01	0.01	0.77	0.06	0.02	0.53	0.45	0.06	0.57	0.30
Our	**0.68**	**0.70**	**0.51**	0.23	0.68	**0.65**	**0.91**	**0.99**	**0.90**	**0.94**	**1.0**	**1.0**	**0.99**	**0.78**	**0.80**

Table 6.4: Accuracies on the TaskAttention dataset. "All" corresponds to the overall accuracy. T1 to T14 correspond to the task categories. T1: sweep floor, T2: mop floor, T3: write on board, T4: clean board, T5: use elevator, T6: pour liquid, T7: make coffee, T8: read book, T9: throw trash, T10: heat food, T11: use computer, T12: search drawer, T13: move bottle, and T14: open door.

6.6 Experiments

6.6.1 Baselines and metric

Baselines. The gaze-related works are most relevant with the attention of a human inside images. Therefore, we take one classic human gaze estimation method and two state-of-the-

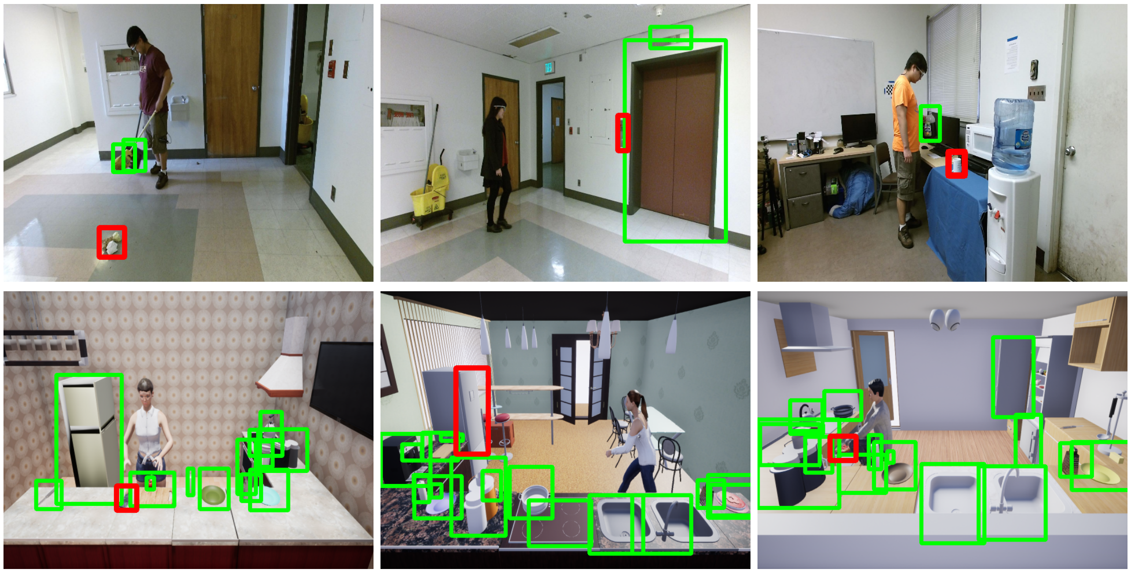


Figure 6.4: Samples of the TaskAttention dataset (upper) and the TaskAttention-VR dataset (lower). The red boxes are the annotations of attentional objects and the green boxes are the annotations of other objects.

art human face and head direction estimation methods as baselines. We briefly introduce the baselines as follows.

-GazeFollow [RKV15] is a gaze estimation method that takes the raw image and human head location as input. The output is the human gaze direction and location.

-PRNet [FWS18] is a face alignment method that can estimate human face direction. It takes the raw image and human face as input, and the output is the dense (more than 40K) aligned face key points. These dense points are compared with a pre-trained model to compute the camera matrix, which is further combined with 68 facial key points to estimate the human face direction.

-Hopenet [RCR18] is a head pose estimation method. It takes the raw image and human face as input, and the output is the three Euler angles that signal human head direction.

Metric. Assume there are n_1 images for testing and human attention in n_2 images is correctly estimated, the metric is the accuracy defined as :

$$acc = \frac{n_2}{n_1} \tag{6.10}$$

Methods	T1	T2	T3	T4	T5	All
GazeFollow [RKV15]	0.37	0.37	0.36	0.29	0.42	0.36
PRNet [FWS18]	0.43	0.43	0.43	0.48	0.43	0.44
Hopenet [RCR18]	0.58	0.60	0.54	0.55	0.55	0.56
Our	**0.77**	**0.77**	**0.79**	**0.72**	**0.70**	**0.74**

Table 6.5: Accuracies on the TaskAttention-VR dataset. "All" corresponds to the overall accuracy. T1 to T5 correspond to the task categories. T1: cook meat, T2: cook soup, T3: make juice, T4: make pizza, and T5: make sandwich.

model	CAD	TaskAttention-VR	TaskAttention
Base	0.66	0.65	0.72
Our	**0.78**	**0.74**	**0.80**

Table 6.6: Overall accuracy comparison with a model named "Base" that takes raw image as input and uses the "Encoder-Decoder+ConvLSTM" neural network architecture.

To evaluate whether an image is correctly estimated, we use the following method. For baselines, the output is gaze/face/head direction, if the direction line intersects with the ground truth attentional objects, the image is counted to be correctly estimated. For our method, to use the same evaluation metric, we first generate a direction line. Let $\mathcal{O} = \{o_k | k = 1, 2, ..., N\}$ be N object candidates and o_i be the object with the highest score that is computed by Eq. 6.9. Let P_o be the center point of o_i, and P_h be the location of human head. The line starting from P_h and passing through P_o is taken as the direction line. Using the same metric, if this direction line intersects with the ground truth attentional objects, the image is counted to be correctly estimated.

To provide the accurate head location for our method and the baselines, we first use the skeleton detector proposed in [CSW17] to detect a human's five key points of nose, left eye, right eye, left ear, and right ear. Then, the average location of the available key points is taken as the head location. To provide the accurate human face for PRNet [FWS18] and Hopenet [RCR18], instead of detecting faces on the whole image, we apply a face detector on a small image region centered on the head location.

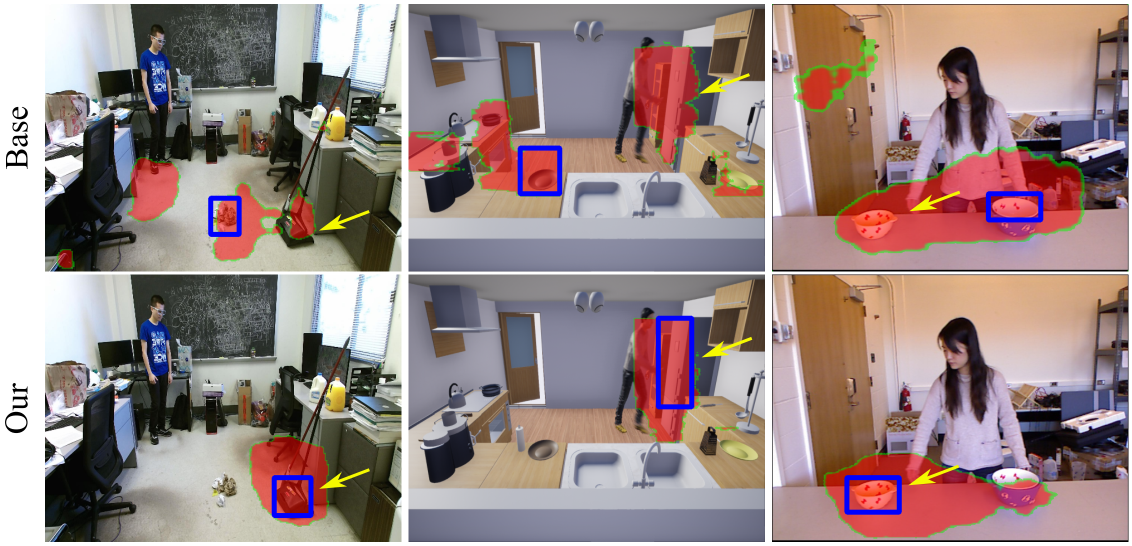


Figure 6.5: Samples of the attention heat map (the red mask) and the most likely attentional object (the blue bounding box) predicted by the "Base" model (upper) and our model (lower). Yellow arrows point to the ground truth attentional objects.

6.6.2 Quantitative results

The quantitative results are summarized in Table. 6.3, Table. 6.4, and Table. 6.5. We can observe that our method outperforms other methods by a large margin. The reasons are two-fold. One one hand, we use the informative low-level human body cues, human skeleton and optical flow, which tend to capture the global motion and pose information, contributing to the good performance when the detail facial information (*e.g.*pupil and eye) is weak. On the other hand, we design a task-driven module to learn the high-level task information. Attention shifting procedure is a task-driven procedure of selecting attentional objects. Therefore, integrating task information with low-level human body cues benefits to accurate attention estimation.

Our method exhibits poor perform in some tasks such as "T1: arranging objects" in the CAD120 dataset and "T4: clean board" in the TaskAttention dataset. After analyzing the results, we found that the majority of objects involved in these tasks are not successfully detected since the training samples of the objects in these tasks are not qualified as other objects.

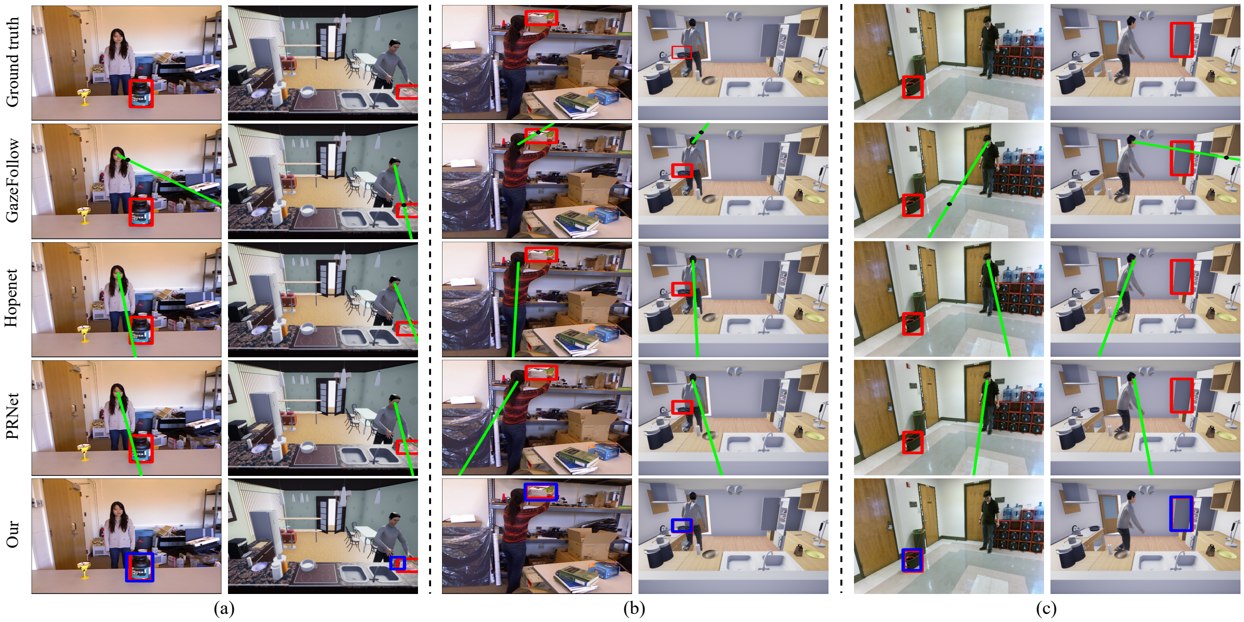


Figure 6.6: Qualitative samples from the baselines and our method in three typical scenarios, (a) the detail human facial information is fully observed, (b) the limited human facial information is available, and c) the human is not gazing at or operating on the attentional objects. In the images, the red bounding boxes are the ground truth attentional objects, the green lines represent the gaze/face/head directions, and the blue bounding boxes are our prediction of attentional objects.

We perform another experiment to further validate the effectiveness of our model. In the experiment, we design a model named as "Base" that uses the "Encoder-Decoder+ConvLSTM" neural network architecture, takes raw image as input, and adopts the same loss function as ours. As shown in Table. 6.6, our model presents the distinct advantage, with 18.2% improvement over the "Base" model on the CAD dataset, 13.8% improvement on the TaskAttention-VR dataset, and 11.1% improvement on the TaskAttention dataset. In Figure 6.5, we show some samples of attention heat map and the attentional object prediction of our model and the "Base" model, from which we can also observe that our model outputs correct attentional object and better attention heat map.

6.6.3 Qualitative results

In some simple scenarios, as shown in Figure 6.6(a), all methods behave well. The reasons are two-fold. On one hand, the humans are gazing at the attentional objects, so the gaze-related methods are effective. On the other hand, the detail information of eyes and faces are fully observed so that the gaze is easy to estimate.

However, in complex cases, as shown in Figure 6.6(b), the humans are not exactly facing to the camera, so the detail facial information is not available. As a result, the methods that heavily depend on facial features present poor performance. When we analyze the experiment results, we found that the most failure of the PRNet [FWS18] and Hopenet [RCR18] happens in these situations. In contrast, instead of using facial features, our model takes human skeleton and optical flow as low-level human body cues. These cues are more robust when the human face is partly or fully occluded, thus presenting better performance in these scenarios.

In some more challenging cases, as shown in Figure 6.6(c), the attentional objects are far from the human, and the human is not gazing at or operating on these objects at the current time. In these cases, the high-level task information is important. For example, as we can see in left column images in Figure 6.6(c), there are a bucket, a trash can, and some bottles in the images. If given the task of mop floor, the attentional object is more likely to be bucket. If given the task of throw trash, the attentional object is more likely to be trash can. If given the task of move bottle, the attentional object is more likely to be bottles. In this work, apart from the pose-motion module considering the low-level human body cues, we design the task-driven module to learn the task information. Therefore, our model behaves better in these cases.

6.7 Conclusion

Human attention can be roughly divided into two categories: 1) the attention of a human outside images, and 2) the attention of a human inside images. Most existing attention-related studies are actually studying the attention of a human outside images. In this work, we are studying the attention of a human inside images, which is related with yet different from human gaze estimation, saliency estimation, human-object interaction detection, and human action detection. The attention of a human inside images is a set of attentional objects that coincide with the human mind and the task a human is doing. As many psychology and biology studies show, human attention is controlled in both bottom-up and top-down manner. Therefore, in this work, we propose a model that considers both low-level human body cues and high-level task information. The experiment results show our model is effective and outperforms other gaze-related methods.

CHAPTER 7

Conclusion

In this dissertation, we have mainly studied the fluents space and their applications in task representation.

Fluents are the time-varying attributes of entities. We first study the objects and their attributes in discrete forms. We extract the feature vector for each image with pre-trained image classification network, we also get the word vector for the object and its attribute from pre-trained words embedding model. We have proposed an encoder-decoder model to map both the image and its description in the same latent space, in which we can recognize new combination of object and attribute which never appears in the training dataset. We have achieve both state-of-art results on two popular datasets, MIT-States [ILA15] and UT-Zappos50K [YG14]. With this model, we can conclude that the generative model is more competitive than discriminative models to recognize unseen classes and the encoder-decoder mechanism is crucial for learning intrinsic fluents representations.

With the idea that describing fluents in image space with latent vector from an encoder-decoder framework. We further study the fluents in a continuous transformation space. In our synthesis face dataset, we have proposed a 4-channel Siamese networks. We utilize a semi-supervised training method with a pair of images as ‘teacher’, we are trying to predict the image which undergoes the same fluents change as teacher’s when given a new image. In this work, we are mainly studying the geometric change which includes both in-plane, out-of-plane, linear and non-linear transformations. The latent space we have learned has the nature of linearity, with which we can manipulate the vector in latent space to decode the image undergoes our desire fluents change.

From the experiment on face datasets, we have demonstrated the mapping of fluents change between image space and latent space through an encoder-decoder generative model. In previous work, fluents change apply to the whole representation of image, which is not interpretable. Hence, on a simulation dataset of objects interaction, we try to disentangle the appearance and geometric fluents, which means we have learned appearance and geometric latent space separately. In the geometric space, we have learned the physics dynamics to predict the position of objects in a long time range.

The process to finish a task is the process to make certain key fluents change step by step. We represent the task with a novel fluent-based task representation framework, in which tasks are modeled with object fluents. In each task, different fluents closely interact with each other by means of spatial concurrence and temporal transition. Given a testing egocentric video, a beam search algorithm is used to jointly recognize the object fluents in each frame, and the task of the entire video. We collect a large-scale egocentric video dataset including various fluents and tasks with detailed annotations. Our experiments have shown that our model outperforms the baseline methods which proves the strength of our model.

Besides objects fluents change in the task, human-object fluent is another important cue in representing a task. Specially, we have studied the human attention change during the task. Human attention can be roughly divided into two categories: 1) the attention of a human outside images, and 2) the attention of a human inside images. Most existing attention-related studies are actually studying the attention of a human outside images. In this work, we are studying the attention of a human inside images, which is related with yet different from human gaze estimation, saliency estimation, human-object interaction detection, and human action detection. The attention of a human inside images is a set of attentional objects that coincide with the human mind and the task a human is doing. As many psychology and biology studies show, human attention is controlled in both bottom-up and top-down manner. We have proposed a model that considers both low-level human body cues and task information. The experiment results show our model is effective and outperforms other gaze-related methods.

This dissertation has provided a glance at the problem of fluents. There still remains many challenges: 1) in spite of the appearance fluents and geometric fluents, topological fluents which can describe more complex changes is much more challenging to model and learn. Topological changes involve the relationship change between entities in the group, which will break of add the relation between entities. 2) To fully represent the task in real life, how to learn a complete fluents space is still unknown. With the complete fluents space landscape, we can discover different paths to finish the task such as with new tools.

REFERENCES

[ARW15] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. "Evaluation of output embeddings for fine-grained image classification." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[ASL17] Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. "Joint discovery of object states and manipulation actions." In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[BAA10] Ali Borji, Majid Nili Ahmadabadi, Babak Nadjar Araabi, and Mandana Hamidi. "Online learning of task-driven object-based visual attention control." *Image and Vision Computing*, 2010.

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2017.

[BPL16] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. "Interaction networks for learning about objects, relations and physics." In *Advances in neural information processing systems (NeurIPS)*, 2016.

[BSI14] Ali Borji, Dicky N Sihite, and Laurent Itti. "What/where to look next? Modeling top-down visual attention in complex interactive environments." *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2014.

[BT06] Neil Bruce and John Tsotsos. "Saliency based on information maximization." In *Advances in neural information processing systems (NeurIPS)*, 2006.

[BT11] William Brendel and Sinisa Todorovic. "Learning spatiotemporal graphs of human activities." In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[CCG16] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. "Synthesized classifiers for zero-shot learning." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CDH16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[CG14] Chao-Yeh Chen and Kristen Grauman. "Inferring analogous attributes." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Che12] Zhe Chen. "Object-based attention: A tutorial review." *Attention, Perception, and Psychophysics*, 2012.

[Cho15] François Chollet et al. "Keras.", 2015.

[CLL15] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems." *arXiv preprint arXiv:1512.01274*, 2015.

[CLL18] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. "Learning to detect human-object interactions." In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[CS02] Maurizio Corbetta and Gordon L Shulman. "Control of goal-directed and stimulus-driven attention in the brain." *Nature reviews neuroscience*, 2002.

[CSV14] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Return of the devil in the details: Delving deep into convolutional nets." *arXiv preprint arXiv:1405.3531*, 2014.

[CSW17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[CY12] W. L. Chou and S. L. Yeh. "Object-based attention occurs regardless of object awareness." *Psychonomic Bulletin and Review*, 2012.

[CZ17] Joao Carreira and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[DDS09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[DPC12] Kun Duan, Devi Parikh, David Crandall, and Kristen Grauman. "Discovering localized attributes for fine-grained recognition." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[EDR94] R Egly, J Driver, and R. D. Rafal. "Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects." *Journal of Experimental Psychology General*, 1994.

[ESE13] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. "Write a classifier: Zero-shot learning using purely textual descriptions." *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[FCS13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. "Devise: A deep visual-semantic embedding model." *Advances in neural information processing systems (NeurIPS)*, 2013.

[FFR11] Alireza Fathi, Ali Farhadi, and James M Rehg. “Understanding egocentric activities.” In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[FGL16] Chelsea Finn, Ian Goodfellow, and Sergey Levine. “Unsupervised Learning for Physical Interaction through Video Prediction.” In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2016.

[FLR12] Alireza Fathi, Yin Li, and James M Rehg. “Learning to recognize daily actions using gaze.” In *European Conference on Computer Vision (ECCV)*, 2012.

[FMO14] Kenneth Alberto Funes Mora, Florent Monay, and Jean-Marc Odobez. “EYEDIAP: A Database for the Development and Evaluation of Gaze Estimation Algorithms from RGB and RGB-D Cameras.” In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications*, 2014.

[FPW16] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. “Spatiotemporal residual networks for video action recognition.” In *Advances in neural information processing systems (NeurIPS)*, 2016.

[FPW17] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. “Spatiotemporal multiplier networks for video action recognition.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[FR13] Alireza Fathi and James M. Rehg. “Modeling actions through state changes.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[FWS18] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. “Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network.” In *European Conference on Computer Vision (ECCV)*, 2018.

[FZ15] Amy Fire and Song-Chun Zhu. “Learning perceptual causality from video.” *ACM Transactions on Intelligent Systems and Technology*, 2015.

[GM15] Saurabh Gupta and Jitendra Malik. “Visual semantic role labeling.” *arXiv preprint arXiv:1505.04474*, 2015.

[Gra12] Alex Graves. “Supervised sequence labelling.” In *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.

[HKW11] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. “Transforming autoencoders.” In *International Conference on Artificial Neural Networks*. Springer, 2011.

[HLV17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. “Densely Connected Convolutional Networks.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[HSB15] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. “Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks.” In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[HT14] Minh Hoai and Fernando De la Torre. "Max-margin early event detectors." *International Journal of Computer Vision (IJCV)*, 2014.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." *IEEE Conference on Computer Vision and pattern recognition (CVPR)*, 2016.

[IK00] Laurent Itti and Christof Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention." *Vision research*, 2000.

[IK01] Laurent Itti and Christof Koch. "Feature combination strategies for saliency-based visual attention systems." *Journal of Electronic imaging*, 2001.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." *IEEE Transactions on pattern analysis and machine intelligence (PAMI)*, 1998.

[ILA15] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. "Discovering States and Transformations in Image Collections." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Itt05] Laurent Itti. "Models of bottom-up attention and saliency." In *Neurobiology of attention*, 2005.

[IZZ17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-To-Image Translation With Conditional Adversarial Networks." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[JED09] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. "Learning to predict where humans look." In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[KAB17] Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. "Deepfix: A fully convolutional neural network for predicting human eye fixations." *IEEE Transactions on Image Processing (TIP)*, 2017.

[KB14] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *CoRR*, **abs/1412.6980**, 2014.

[KBN08] Neeraj Kumar, Peter Belhumeur, and Shree Nayar. "Facetracer: A search engine for large collections of images with faces." *European Conference on Computer Vision (ECCV)*, 2008.

[KCS17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. "The kinetics human action video dataset." *arXiv preprint arXiv:1705.06950*, 2017.

[KF18] Yu Kong and Yun Fu. "Human Action Recognition and Prediction: A Survey." *arXiv preprint arXiv:1806.11230*, 2018.

[KGS13] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. "Learning human activities and object affordances from rgb-d videos." *The International Journal of Robotics Research*, 2013.

[KPO13] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. "Babytalk: Understanding and generating simple image descriptions." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[KTB14] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet." *arXiv preprint arXiv:1411.1045*, 2014.

[KTS14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[KU87] Christof Koch and Shimon Ullman. "Shifts in selective visual attention: towards the underlying neural circuitry." In *Matters of intelligence*, 1987.

[KWK15] Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. "Deep Convolutional Inverse Graphics Network." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[KXG17] Elyor Kodirov, Tao Xiang, and Shaogang Gong. "Semantic autoencoder for zero-shot learning." *arXiv preprint arXiv:1704.08345*, 2017.

[Lap05] Ivan Laptev. "On space-time interest points." *International Journal of Computer Vision (IJCV)*, 2005.

[LGF16] Adam Lerer, Sam Gross, and Rob Fergus. "Learning Physical Intuition of Block Towers by Example." In *International Conference on Machine Learning (ICML)*. PMLR, 2016.

[LGG17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[LHK14] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. "The secrets of salient object segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[LKZ17] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Schmidt Feris. "Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[LMS08] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. "Learning realistic human actions from movies." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[LNH14] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. "Attribute-based classification for zero-shot visual object categorization." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[LRT14] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. "Transient attributes for high-level understanding and editing of outdoor scenes." *ACM Transactions on Graphics*, 2014.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[LSF15] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. "Predicting deep zero-shot convolutional neural networks using textual descriptions." *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[LWZ17] Yang Liu, Ping Wei, and Song-Chun Zhu. "Jointly Recognizing Object Fluents and Tasks in Egocentric Videos." *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[LYR15] Yin Li, Zhefan Ye, and James M Rehg. "Delving into egocentric actions." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[MCL15] Michal Mathieu, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error." *CoRR*, **abs/1511.05440**, 2015.

[MFK16] Minghuang Ma, Haoqi Fan, and Kris M Kitani. "Going deeper into first-person activity recognition." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[MGH17] Ishan Misra, Abhinav Gupta, and Martial Hebert. "From red wine to red tomato: Composition with context." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[MH08] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of Machine Learning Research (JMLR)*, 2008.

[MK03] Notger G Müller and Andreas Kleinschmidt. "Dynamic interaction of object-and space-based attention in retinotopic visual areas." *Journal of Neuroscience*, 2003.

[ML16] Arun Mallya and Svetlana Lazebnik. "Learning models for actions and person-object interactions with transfer to question answering." In *European Conference on Computer Vision (ECCV)*, 2016.

[MLS09] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. "Actions in context." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[MSN11] Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. “A joint learning framework for attribute models and object descriptions.” *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[MSS16] Shugao Ma, Leonid Sigal, and Stan Sclaroff. “Learning activity progression in LSTMs for activity detection and early detection.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[MTV06] A. Martínez, W. Teder-Sälejärvi, M. Vazquez, S. Molholm, J. J. Foxe, D. C. Javitt, F. Di Russo, M. S. Worden, and S. A. Hillyard. “Objects Are Highlighted by Spatial Attention.” *Journal of Cognitive Neuroscience*, 2006.

[Mue15] Erik T. Mueller. *Commonsense Reasoning: An Event Calculus Based Approach.* Morgan Kaufmann Publishers Inc., 2 edition, 2015.

[NG18] Tushar Nagarajan and Kristen Grauman. “Attributes as Operators.” *European Conference on Computer Vision (ECCV)*, 2018.

[NMB13] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. “Zero-shot learning by convex combination of semantic embeddings.” *arXiv preprint arXiv:1312.5650*, 2013.

[OGL15] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. “Action-conditional video prediction using deep networks in atari games.” In *Advances in neural information processing systems (NeurIPS)*, 2015.

[PBI15] Daniel Parks, Ali Borji, and Laurent Itti. “Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes.” *Vision research*, 2015.

[PG11] Devi Parikh and Kristen Grauman. “Relative attributes.” *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[PH12] Genevieve Patterson and James Hays. “Sun attribute database: Discovering, annotating, and recognizing scene attributes.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[PKA09] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. “A 3D Face Model for Pose and Illumination Invariant Face Recognition.” In *IEEE*, 2009.

[PR12] Hamed Pirsiavash and Deva Ramanan. “Detecting activities of daily living in first-person camera views.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[PR14] A Pooresmaeili and P. R. Roelfsema. “A Growth-Cone Model for the Spread of Object-Based Attention during Contour Grouping.” *Current Biology*, 2014.

[PSH18] Seonwook Park, Adrian Spurr, and Otmar Hilliges. “Deep Pictorial Gaze Estimation.” *arXiv preprint arXiv:1807.10002*, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation." *Empirical Methods in Natural Language Processing*, 2014.

[PYY17] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. "Transformation-Grounded Image Generation Network for Novel 3D View Synthesis." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[PZB18] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. "Learning to find eye region landmarks for remote gaze estimation in unconstrained settings." *arXiv preprint arXiv:1805.04771*, 2018.

[QWJ18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. "Learning Human-Object Interactions by Graph Parsing Neural Networks." In *European Conference on Computer Vision (ECCV)*, 2018.

[RA09] Michael S Ryoo and Jake K Aggarwal. "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities." In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[RCR18] Nataniel Ruiz, Eunji Chong, and James M. Rehg. "Fine-Grained Head Pose Estimation Without Keypoints." In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[RDS14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. "ImageNet Large Scale Visual Recognition Challenge." *CoRR*, **abs/1409.0575**, 2014.

[RDS15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision (IJCV)*, 2015.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with Region Proposal Networks." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RKV15] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. "Where are they looking?" In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RVK17] Adria Recasens, Carl Vondrick, Aditya Khosla, and Antonio Torralba. "Following Gaze in Video." In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[Ryo11] Michael S Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos." In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[SAJ16] Suriya Singh, Chetan Arora, and CV Jawahar. "First person action recognition using deep learned descriptors." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SB06] Sarah Shomstein and Marlene Behrmann. "Cortical systems mediating visual attention to both objects and spatial locations." *PNAS*, 2006.

[SC94] Barry Smith and Roberto Casati. "Naive physics." *Philosophical Psychology*, 1994.

[Sch01] Brian J Scholl. "Objects and attention: the state of the art." *Cognition*, 2001.

[SDN08] Silvio Savarese, Andrey DelPozo, Juan Carlos Niebles, and Li Fei-Fei. "Spatial-temporal correlatons for unsupervised action classification." In *IEEE Workshop on Motion and Video Computing*, 2008.

[See11] Axel Seemann. *Joint attention: New developments in psychology, philosophy of mind, and social neuroscience.* MIT Press, 2011.

[SFC11] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. "Real-time human pose recognition in parts from single depth images." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[SGM13] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. "Zero-shot learning through cross-modal transfer." *Advances in neural information processing systems (NeurIPS)*, 2013.

[SKB12] Walter J Scheirer, Neeraj Kumar, Peter N Belhumeur, and Terrance E Boult. "Multi-attribute spaces: Calibration for attribute fusion and similarity search." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[SL16] Krishna Kumar Singh and Yong Jae Lee. "End-to-end localization and ranking for relative attributes." *European Conference on Computer Vision (ECCV)*, 2016.

[SLC04] Christian Schuldt, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." In *International Conference on Pattern Recognition (ICPR*, 2004.

[SMS15] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. "Unsupervised learning of video representations using lstms." In *International Conference on Machine Learning (ICML)*, 2015.

[SPS12] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. "Unstructured human activity detection from rgbd images." In *IEEE international conference on robotics and automation (ICRA)*, 2012.

[SPY11] Zhangzhang Si, Mingtao Pei, Benjamin Yao, and Song-Chun Zhu. "Unsupervised learning of event and-or grammar and semantics from video." In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[SYF13] Brian A Smith, Qi Yin, Steven K Feiner, and Shree K Nayar. "Gaze locking: passive eye contact detection for human-object interaction." In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013.

[SZ14a] Karen Simonyan and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[SZ14b] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[SZX16] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. "Deep attributes driven multi-camera person re-identification." *European Conference on Computer Vision (ECCV)*, 2016.

[VAM18] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. "Generalized zero-shot learning via synthesized examples." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[WBM16] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. "A 3d morphable eye region model for gaze estimation." In *European Conference on Computer Vision (ECCV)*, 2016.

[WBS04] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. "Image quality assessment: from error visibility to structural similarity." *IEEE Transactions on Image Processing (TIP)*, 2004.

[WCC16] Wenlin Wang, Changyou Chen, Wenlin Chen, Piyush Rai, and Lawrence Carin. "Deep metric learning with data summarization." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016.

[WCK16] William F. Whitney, Michael Chang, Tejas Kulkarni, and Joshua B. Tenenbaum. "Understanding Visual Concepts with Continuation Learning." *CoRR*, **abs/1502.04623**, 2016.

[WFG16] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. "Actions ˜ transformations." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[WG18] Xiaolong Wang and Abhinav Gupta. "Videos as Space-Time Region Graphs." In *European Conference on Computer Vision (ECCV)*, 2018.

[WGT17] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. "Interpretable Transformations With Encoder-Decoder Networks." In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[WJ13] Xiaoyang Wang and Qiang Ji. “A unified probabilistic approach modeling relationships between attributes and objects.” *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[WJ17] Kang Wang and Qiang Ji. “Real time eye gaze tracking with 3d deformable eyeface model.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[WJW13] Shuo Wang, Jungseock Joo, Yizhou Wang, and Song-Chun Zhu. “Weakly supervised learning for attribute localization in outdoor scenes.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[WKS11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. “Action recognition by dense trajectories.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[WLK17] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. “Learning to See Physics via Visual De-animation.” In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2017.

[WLS18] Ping Wei, Yang Liu, Tianmin Shu, Nanning Zheng, and Song-Chun Zhu. “Where and why are they looking? jointly inferring human attention and intentions in complex tasks.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[WM10] Yang Wang and Greg Mori. “A discriminative latent model of object classes and attributes.” *European Conference on Computer Vision (ECCV)*, 2010.

[WPV18] Wenlin Wang, Yunchen Pu, Vinay Kumar Verma, Kai Fan, Yizhe Zhang, Changyou Chen, Piyush Rai, and Lawrence Carin. “Zero-shot learning via class-conditioned deep generative models.” *AAAI Conference on Artificial Intelligence*, 2018.

[WQT15] Limin Wang, Yu Qiao, and Xiaoou Tang. “Action recognition with trajectory-pooled deep-convolutional descriptors.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[WS13] Heng Wang and Cordelia Schmid. “Action recognition with improved trajectories.” In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[WS18] Wenguan Wang and Jianbing Shen. “Deep visual attention prediction.” *IEEE Transactions on Image Processing (TIP)*, 2018.

[WSS18] Wenguan Wang, Jianbing Shen, and Ling Shao. “Video salient object detection via fully convolutional networks.” *IEEE Transactions on Image Processing (TIP)*, 2018.

[WSW17] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. "Image captioning and visual question answering based on attributes and external knowledge." *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2017.

[WVJ16] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. "How to Use t-SNE Effectively." *Distill*, 2016.

[WXZ17] Ping Wei, Dan Xie, Nanning Zheng, and Song-Chun Zhu. "Inferring Human Attention by Learning Latent Intentions." In *IJCAI*, 2017.

[WYL15] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. "Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning." In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2015.

[WZW17] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. "Visual interaction networks: Learning a physics simulator from video." In *Advances in neural information processing systems (NeurIPS)*, 2017.

[WZX16] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[WZZ13a] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4d human-object interactions for event and object recognition." In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[WZZ13b] Ping Wei, Nanning Zheng, Yibiao Zhao, and Song-Chun Zhu. "Concurrent action detection with structural prediction." In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[WZZ17] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4D human-object interactions for joint event segmentation, recognition, and object Localization." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[XCW15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In *Advances in neural information processing systems (NeurIPS)*, 2015.

[XLL16] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. "Multi-view people tracking via hierarchical trajectory composition." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[XLW18] Bingjie Xu, Junnan Li, Yongkang Wong, Mohan S Kankanhalli, and Qi Zhao. "Interact as You Intend: Intention-Driven Human-Object Interaction Detection." *arXiv preprint arXiv:1808.09796*, 2018.

[YG14] Aron Yu and Kristen Grauman. "Fine-grained visual comparisons with local learning." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[YHV15] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond short snippets: Deep networks for video classification." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[YRY15] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[YYS16] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. "Attribute2image: Conditional image generation from visual attributes." *European Conference on Computer Vision (ECCV)*, 2016.

[YZL13] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. "Saliency detection via graph-based manifold ranking." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[ZB16] Yipin Zhou and Tamara L. Berg. "Learning Temporal Transformations From Time-Lapse Videos." In *European Conference on Computer Vision (ECCV)*, 2016.

[ZDG14] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. "Part-based R-CNNs for fine-grained category detection." In *European Conference on Computer Vision (ECCV)*, 2014.

[ZMJ17] Xilin Zhang, Nicole Mlynaryk, Shruti Japee, and Leslie G Ungerleider. "Attentional selection of multiple objects in the human visual system." *NeuroImage*, 2017.

[ZSF15] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. "Appearance-based gaze estimation in the wild." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[ZSF17a] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. "It's written all over your face: Full-face appearance-based gaze estimation." In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[ZSF17b] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation." *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2017.

[ZTS16] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. "View Synthesis by Appearance Flow." In *European Conference on Computer Vision (ECCV)*, 2016.

[ZZW16] Xiaochun Zou, Xinbo Zhao, Jian Wang, and Yongjia Yang. "Learning to model task-oriented attention." *Computational intelligence and neuroscience*, 2016.