UNIVERSITY OF CALIFORNIA SAN DIEGO

Unaligned Sequential Image Transformation with GAN

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Siyang Wang

Committee in charge:

Professor Zhuowen Tu, Chair
Professor David Kriegman
Professor Lawrence Saul

2019

The thesis of Siyang Wang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Zhuowen Tu for his support as the chair of my committee. Through multiple drafts and many long nights, his guidance has proved to be invaluable.

This entire thesis, in part is currently being prepared for submission for publication of the material. Siyang Wang; Justin Lazarow; Kwonjoon Lee; Zhuowen Tu. The thesis author was the primary investiator and author of this material. Thanks to my co-authors.

I would also like to thank Weijian Xu for giving me countless helpful discussions.

ABSTRACT OF THE THESIS

Unaligned Sequential Image Transformation with GAN

by

Siyang Wang

Master of Science in Computer Science

University of California San Diego, 2019

Professor Zhuowen Tu, Chair

The purpose of this thesis is to present a novel method of learning to generate an image sequence from input a single image without sequentially aligned data. Given examples of a visual phenomena that can be divided into discrete time steps, the problem is to learn a model that takes an input from any such time and realizes this input at all other time steps in the sequence. For example, given a scenery picture in spring and output the corresponding pictures in sequence of summer, fall, and winter without changing overall layout and semantic information presented in the input picture. Furthermore, it is assumed that ground-truth aligned sequences are not provided. This broadens the real-world application of this method as it is often difficult to collect aligned sequential data for many problems. This task generalizes the unpaired image-to-image

problem from generating pairs to generating sequences and associates a direction of time with the phenomena observed.

We show that this problem can be solved by incorporating Generative Adversarial Networks (GAN), a popular deep unsupervised learning technique, and a periodic assumption about the sequential visual phenomena modeled. The periodic assumption is enforced in model training by a novel Loop Consistency loss, inspired by the popular Cycle Consistency loss that has achieved huge success in unpaired image-to-image transformation. The two parts of the model, GAN and Loop Consistency, can be seen as two levels of constraints that together facilitate model training. The transformation unit itself is a neural network. We show the effects of different network architecture changes on generation quality and present the results of the model in comparison with several competitive baseline models.

Chapter 1

Introduction

Image-to-image translation is the problem of learning a transformation function that takes input from one image domain and transform it into another image domain that is both realistic and semantically meaningful. There are many examples of this problem in application. Given a young person's face, we'd like to learn a function that transforms it into an old face without changing the overall facial structure and features. Another example would be learning a function that transforms a scenery picture in spring to winter without changing the layout and semantic components, such as trees, mountains, creeks present in the picture. In this specific example, the learned transformation function should have some level of understanding about the semantic components and their corresponding transformation into the other domain, such as that the trees should be covered in snow and the creeks should be frozen when transforming from spring to winter.

Machine learning approaches have dominated this task in recent years and will be the focus of this thesis. If aligned data is available, then this is a regression problem. As an example, in order to learn a transformation from young faces to old faces, first collect an aligned image data set consisting of the same person's face image at both a young and an old age. In mathematical terms, this means that we have a data set with x_i's and y_i's such that there's a ground truth function

$f(x_i) = y_i \; \forall i$, and the problem is to approximate f. The straightforward regression approach then tries to minimize some loss function $L = ||G(x_i) - y_i||_1$ with respect to some hypothesis G. The problem is that using a simple distance function as the loss (l_1 distance in the previous example) does not capture well the closeness of the generated image $G(x_i)$ and the ground truth y_i. In the face aging example, if $G(x_i) = y_i^*$ where y_i^* is y_i but translated to the left by some pixels, then $L = ||G(x_i) - y_i||_1 = ||y_i^* - y_i||_1$ would be big. This is clearly not useful for training G as the loss function is still giving error signal despite G already generating good results.

Generative Adversarial Networks (GAN) [GPAM$^+$14] is an unsupervised learning approach that learns a data distribution by jointly training a generator and a discriminator. The generator maps a noise vector (usually drawn from a multi-dimensional Gaussian) to the target data distribution space. The generator's goal is to fool the discriminator whose job is to distinguish generated data from real data. The loss for the generator function in GAN (referred as GAN loss) is the loss of the discriminator function accuracy (often calculated with cross entropy), thus the discriminator can be seen as a learned loss function that is better than a vanilla distance function. In image problems, the discriminator is usually a Convolutional Neural Network (CNN) [LBB$^+$98] which is able to extract low-level and high-level features that are translation-invariant. It is important to note that the generator function G is also a CNN in unsupervised learning of image distributions.

A pioneering work, pix2pix [IZZE17] shows that it is possible to realize a real image from one domain as a highly realistic and semantically meaningful image in another with aligned data set using a combination of distance function loss (such as l_1) and GAN loss. This is the current state-of-the-art model in image-to-image translation with aligned data. However, data alignment is usually difficult to obtain for many image-to-image translation problems. For example, in scenery picture season change problem, it is very difficult to take a picture at the same location in spring and winter, not to mention doing this for many different locations. In other cases, a ground-truth alignment may not exist at all, such as transforming a zebra to a horse. A meaningful

transformation can still be made by painting the body of the horse with the color pattern of the zebra, but it's impossible to collect aligned data.

The authors of pix2pix [IZZE17] successfully extended their GAN-based framework that requires aligned data to unaligned image-to-image transformation by adding another layer of constraint that they call *Cycle Consistency* [ZPIE17] to compensate for aligned data. Their proposed model CycleGAN trains a pair of generators, $G_{A->B}$ and $G_{B->A}$, between two image domains X_A and X_B and two discriminators in each domain, D_A and D_B. They require the generators to not only fool the discriminators but also conform to *Cycle Consistency* $||G_{B->A}(G_{A->B}(x_A)) - x_A||_1$ where $x_A \sim X_A$. Methods [KCK$^+$17, LBK17] similar to CycleGAN have also been developed roughly around the same time. These methods are state-of-the-art in many unaligned image-to-image problems. The key to the success of these methods including CycleGAN is the combination of GAN loss and an added layer of constraint such as *Cycle Consistency*.

However, CycleGAN family models are still somewhat limited since they only handle the translation problem (directly) between two domains. Modeling more than two domains would require separate instantiations of CycleGAN between any two pairs of domains — resulting in a quadratic model complexity. A major recent work, StarGAN [CCK$^+$18], addresses this by facilitating a fully connected domain-translation graph, allowing transformation between two arbitrary domains with a single model. This flexibility, however, appears restricted to domains corresponding to specific attribute changes such as emotions and appearance in face image transformation. It is designed for general-purpose multi-domain transformation without aligned data and does not directly model the sequential multi-domain problems that we are trying to solve.

Within nature, a multitude of settings exist where neither a set of pairs nor a fully-connected graph are the most natural representations of how one might proceed from one domain to another. In particular, many natural processes are sequential or even periodic, often with a notion of time, and therefore the translation process should reflect this. A common phenomena

modeled as an image-to-image task is the visual change of natural scenes between two seasons [ZPIE17], , Winter and Summer. This neglects the fact that nature first proceeds to Spring after Winter and Fall after Summer and therefore the pairing induces a very discontinuous reflection of the underlying process. Instead, we hope that by modeling a higher resolution discretization of this process, the model can more realistically approach the true model while reducing the necessary complexity of the model.

Aligned sequences of a process are often even more difficult to come by than paired image-to-image data. It is already very difficult to collect the face image of the same person at two different ages which means doing so for more than two ages is even more difficult. Thus, it is more plausible to gather a large number of examples from each step (domain) in a sequence without correspondences between the content of the examples. It then becomes crucial to mimic the success of unpaired image-to-image transformation. Therefore, we consider a setting where the process being modeled is periodic (or can be made so) with unaligned examples from each time step. Given an example from an arbitrary point in the sequence, we then generate an *aligned* sequence over all other time steps — expecting a faithful realization of the image at each step. The key condition required is that after generating an entire loop (returning from the last domain to the input domain), one should expect to return to the original input. This is quite a weak condition and promotes model flexibility, because at any step within the loop, the transformation can be very drastic as long as the final step of the loop returns to the original input. We denote this extension to the *Cycle Consistency* of [ZPIE17] as *Loop Consistency* and therefore name our approach as *Loop-Consistent Generative Adversarial Networks (LoopGAN)*. This is a departure from many image-to-image approaches that have very short (usually length 2) paths of computation defining what it means to have gone "there and back", e.g. the ability to enforce reconstruction or consistency. Since we do not have aligned sequences, the lengths of these paths for LoopGAN are as large as the number of domains being modeled and require different approaches to make learning feasible. These are not entirely different from the problems

that often arise in recurrent neural networks and we can draw similarities to our model as a memory-less recurrent structure applied to images.

We apply our method to the sequential (although aperiodic) phenomena of human aging [ZQ17], as well as, the periodic phenomena of the seasons of the Alps [AATVG18]. We build three sets of baseline models. The first set consists of popular unpaired image-to-image models CycleGAN [ZPIE17] and UNIT [LBK17]. These two models are chosen because they represent two main families of unpaired image-to-image models that differ in the way that *Cycle Consistency* is enforced. CycleGAN represents a family of models [LTH+18] [ARS+18] that *explicitly* enforces *Cycle Consistency*, while UNIT represents a family of models [LBK17] [HLBK18] that *implicitly* enforces *Cycle Consistency*. Because CycleGAN and UNIT can only be applied to two domains at a time, we train a separate model for each pair of domains in the sequential domains of the two data sets. For example, in changing seasons problem, we train a separate model between every two seasons which amount to 6 models in total to facilitate transformation between the four domains (seasons). Besides this radical approach to achieve sequential generation with bi-domain models, we are also interested in the ability of bi-domain models learning sequential generation. We thus build another set of baseline models by chaining bi-domain models between consecutive domains in a sequential transformation problem. Take changing seasons problem as an example again. We train a CycleGAN between consecutive seasons as spring-summer, summer-fall, fall-winter, winter-spring, four in total, two less than training a model between every two seasons. We do the same for UNIT. The third set of baseline models consists of just the aforementioned StarGAN [CCK+18], a general-purpose multi-domain model. This is to show the difference between a general-purpose multi-domain model and ours which explicitly models a multi-domain problem that has a natural sequential order. We show favorable results against all three sets of baseline methods in two sequential image transformation tasks.

Chapter 2

GAN Approaches to Image-to-image Transformation

This chapter provides the main context for the thesis which is image-to-image transformation approaches that incorporate GAN. The foundations, GAN and DCGAN, are introduced first. The application of GAN in image-to-image transformation is introduced afterwards. In the last part, some related areas that are not necessarily GAN or image-to-image transformation are briefly introduced to provide full context for the thesis' proposed method.

2.1 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN) is first proposed in [GPAM$^+$14] as an unsupervised approach of learning a data distribution. The key is to simultaneously train a pair of generator and discriminator such that the generator is trying to generate data points to fool the discriminator which is tasked to distinguish between generated (fake) and real data points. The GAN framework has the unique advantage of approximating a data distribution with a learned distance function, i.e. the discriminator. This is especially useful for high-dimensional data such

as images, as existing mathematical distance measures such as $l-1$ or $l-2$ are too rigid for image data and do not capture the true closeness of two image data distributions. For example, translating the same image to the left by a few pixels would result in a very high $l-1$ distance while the two images are essentially the same. In the GAN framework, however, the discriminator can be designed to especially attend to the characteristics of the data distribution. In the case of images, the discriminator is usually a CNN which is able to extract both low-level and high-level features while being translation-invariant. Thus, a CNN-based discriminator can approximate the closeness of real data distribution and generator output much better than a traditional distance measure such as $l-1$ and better guides the generator to generate high-quality images, i.e. better learning the underlying distribution.

2.2 Deep Convolutional GAN (DCGAN)

A class of GAN designed to learn image data distribution is Deep Convolutional GAN (DCGAN) [RMC15]. This model uses CNN architecture for both discriminator and generator. The discriminator has similar architecture as a normal image classification CNN. The generator has a special type of convolution operation that is not often used in CNN context known as deconvolution. It can be seen as the reverse operation of convolution. The normal convolution operations used in classification CNN produce smaller feature maps. Deconvolution, on the other hand, produces feature maps that are bigger than the input. If we see convolution as a form of downsampling, then deconvolution is a form of upsampling. Deconvolution is key to DCGAN generator, because the input to the generator is a noise vector that is low in dimension compared to the output. Layers of deconvolution magnify the low-dimensional input noise signal to high-dimensional image output. DCGAN extends the frame work of GAN to image data and shows that incorporating CNN with GAN is the state-of-the-art approach to learn image data distribution in an unsupervised manner.

2.3 Supervised Image-to-image Transformation with GAN

Image-to-image transformation is the task of transforming an image in one domain to another. Here, domains are loosely defined as a set of images with shared semantic meaning and appearance similarity. They are also sometimes defined according to the transformation task. Face aging is a popular image-to-image transformation problem. Define two image domains X_A and X_B as young and old. The problem is to learn a transformation function that takes in an image from X_A and transform it into X_B. However, there are usually additional constraints on the transformed image, otherwise the function can just output some existing image in X_B. It is assumed that there exists some ground-truth transformation. In a data set that is paired (or sometimes called supervised), this ground-truth transformation is partially represented in image pairing. Thus, the problem can be easily framed as a regression problem. We simply regress $G_{A->B}$ to ground-truth by minimizing $E_{(x_A,x_B)\ (X_A,X_B)}||G_{A->B}(x_A)-x_B||_1$ where (X_A,X_B) is the paired data distribution.

However, the problem with regressing to a simple distance function is that it can be too rigid in some cases. The same image translated by a few pixels results in a high $l-1$. Thus, it clearly does not represent the true distance well. Adding GAN to the loss function alleviates this problem as the distance function is now learned and can guide the generator (transformation function) to match both low-level and high-level features of the target transformation domain distribution. It's important to note that the type of GAN used in image-to-image transformation is generally called conditional GAN (cGAN) as the generator is conditioned on a specific prior (an image in this case) instead of a noise function.

It is shown [IZZE17] that this approach produces state-of-the-art transformation when paired data is available. In the ablation studies, the authors showed that using $l-1$ regression by itself often causes the produced image to be blurry. They suggested that the generator is afraid of producing sharp images because any pixel-level difference would cause high $l-1$ error and is

thus encouraged to produce blurry images that capture the overall structure and have relatively low $l-1$ error. The final version of their model pix2pix combines $l-1$ with GAN and produces better images than either loss function by itself.

2.4 Unsupervised Image-to-image Transformation

Paired image-to-image data can be hard to collect. For example, it would be very difficult to collect a large number of scenery pictures taken at the exactly the same location and angle in two different seasons (spring and winter). It is more feasible to collect a large number of spring scenery picture and winter scenery pictures without paired relationship in the sense that no two pictures are taken at exactly the same location and angle. How then to learn a transformation between two image domains without paired relationship?

An extension to the work on pix2pix [IZZE17], Zhu *et al.* proposed CycleGAN [ZPIE17] which has a similar GAN setup as [IZZE17] but is able to learn transformation between two domains without paired training data. To achieve this, CycleGAN simultaneously train two generators, one for each direction between the two domains. Besides the GAN loss enforced upon by domain-wise discriminators, the authors proposed to add a cycle-consistency loss which forces the two generators to be reversible. This consistency loss turns out to be the key as in the ablation study shows that the model breaks and produces incoherent images with just GAN loss [Zhu19].

Concurrent to CycleGAN, UNIT [LBK17] proposed a method that implicitly enforces cycle consistency using a VAE-like structure while achieving comparable results. It assumes that the two image domains share a latent space. The model trains two sets of encoder-decoder in each domain. The loss consists of two parts, VAE loss and implicit cycle-consistency loss. The VAE loss guides the in-domain encoder-decoder to learn the in-domain image distribution. The cycle-consistency loss guides a cross-domain encoder-decoder pair to first encode an image in the source domain to the shared-latent space and then decode into the target domain where a

discriminator is trained to distinguish real and transformed images. The VAE loss is $l-1$ and the cycle-consistency loss is GAN. The cycle consistency is not directly enforced as in CycleGAN but through the shared-latent space assumption.

A major problem with both CycleGAN and UNIT is that the learned transformation has to be deterministic or one-to-one. This assumption is not true for many image-to-image problems. For example, in transforming a cat picture to leopard, there exists many plausible transformations. Restricting the scope of transformation to singular could also potentially cripple the training process as the model tries hard to find the one transformation while the data presents a variety of possibilities that confuse the model. To solve this problem, one-to-many non-deterministic unsupervised models [HLBK18] [ARS$^+$18] are proposed. Augmented CycleGAN [ARS$^+$18] is proposed as an extension to vanilla CycleGAN that is enabled to make one-to-many transformations by condition the transformation on both the input image and a noise vector. Multi-modal transformation [HLBK18] is proposed as an extension to UNIT to achieve one-to-many transformation in a VAE-like setting. This model proposes to learn two separate latent spaces, one is shared between the two image domains called content code, the other is domain specific called style code. A transformation is made by first embed the input image into the content code space and then sample a style code from the target domain style code distribution and then combine the two as input to a decoder to the target domain.

Both CycleGAN family and UNIT family models have achieved great success in image-to-image transformation without paired training data. However, they are not designed to be directly applied to problems with more than two domains. For example, transforming a spring image to not just winter but also summer and fall. Two simple solutions would be to either train all pair-wise reversible transformations A-B, A-C, A-D, B-C, B-D, C-D (if the domains are A,B,C,D) or to train minimum number of pair-wise reversible transformations that bridge domains in a chained fashion, A-B, B-C, C-D, D-A. Note that in order to transform from A to C in the chained approach, the image needs to be first transformed into B trough A-B and then to C through B-C.

Either approach requires the number of parameters proportional to the number of domains. These are the baseline models we compare our model against.

2.5 Multi-domain Unsupervised Transformation

StarGAN [HLBK18] extends CycleGAN for any number of domains. A single network takes inputs defining the source image and desired domain transformation. This model is aimed at general-purpose multi-domain unsupervised transformation, meaning that it makes no assumption about the relationship between the domains, such as temporal relationship. However, it is has been mainly shown to be successful for the domains consisting of facial attributes and expressions of emotion.

The generality of StarGAN's application scope could also be its pitfall in some specific applications. We argue through experiments that it has limited usage in multi-domain problems where there exists a ground-truth temporal relationship between domains.

2.6 Other Related Works

2.6.1 Style Transfer

A specific task in image-to-image transformation called style transfer is broadly defined as the task of transforming a photo into an artistic style while preserving its content [GEB15, JAFF16]. Common approaches use a pre-trained CNN as feature extractor and optimize the output image to match low-level features with that of style image and match high-level features with that of content image [GEB15, JAFF16]. A network architecture innovation made popular by this field known as AdaIn [HB17, DSK17] combines instance normalization with learned affine parameters. It needs just a small set of parameters compared to the main network weights achieve different style transfers within the same network. It also shows great potential in improving image

quality for image generation [KLA19] and image-to-image transformation [HLBK18]. This is a key component of the network architecture proposed in this work.

2.6.2 Face Aging

Generating a series of faces in different ages given a single face image has been widely studied in computer vision. State-of-the-art methods [ZQ17, PATVG18] use a combination of pre-trained age estimator and GAN to learn to transform the given image to different ages that are both age-accurate and preserve original facial structure. They rely heavily on a domain-specific age estimator and thus have limited application to the more general sequential image generation tasks that we try to tackle here.

2.6.3 Video Prediction

Video prediction attempts to predict some number of future frames of a video based on a set of input frames [SCW$^+$15, VPT16]. Full videos with annotated input frames and target frames are often required for training these models. A combination of RNN and CNN models has seen success in this task [SMS15, SCW$^+$15]. Predictive vision techniques [VPT16, VT17, WJY$^+$19] that use CNN or RNN to generate future videos also require aligned video clips in training. A recent work [GJFF$^+$18] added a GAN as an extra layer of supervision for learning human trajectories. At a high level, video prediction can be seen as a supervised setting of our unsupervised task. Moreover, video prediction mostly aims at predicting movement of objects rather than transformation of a still object or scene which is the focus of our task.

Chapter 3

Unaligned Sequential Transformation with Loop-Consistent GAN

This chapter proposes a GAN-based sequential image transformation model. The overall model formulation, loss objectives, and detailed network architectures are introduced followed by comparisons with several baseline models on two sequential image transformation tasks. At the end of the chapter, some aspects of the model are further explored with extended experiments.

3.1 Method

We formulate our method and objectives for a GAN-based sequential image transformation model in this section. Consider a setting of n domains, $X_1, \ldots, X_n$ where $i < j$ implies that X_i occurs temporally before X_j. This defines a sequence of domains. To make this independent of the starting domain, we additionally expect that can translate from X_n to X_1 — something a priori when the sequence represents a periodic phenomena. We define a *single* generator $G(x, i)$ where $i \in \{1, \ldots, n\}$ and $x \in X_i$. Then, a translation between two domains X_i and X_j of an input $x_i \in X_i$ is given by repeated applications of G in the form of $G^{\|j-i\|}(x_i, i)$ (allowing for incrementing

the second argument modulo $n+1$ after each application of G). By applying G to an input n times, we have formed a direct loop of translations where the source and target domains are equal. While we use a single generator, we make use of n discriminators $\{D_i\}_{i=1}^{n}$ where D_i is tasked with discriminating between a translation from any source domain to X_i. Since we are given only samples from each domain X_i, we refer to each domain $X_i = \{x_j\}_{j=1}^{N_i}$ as consisting of N_i examples from the domain X_i with data distribution $p_{data}(x_i)$.

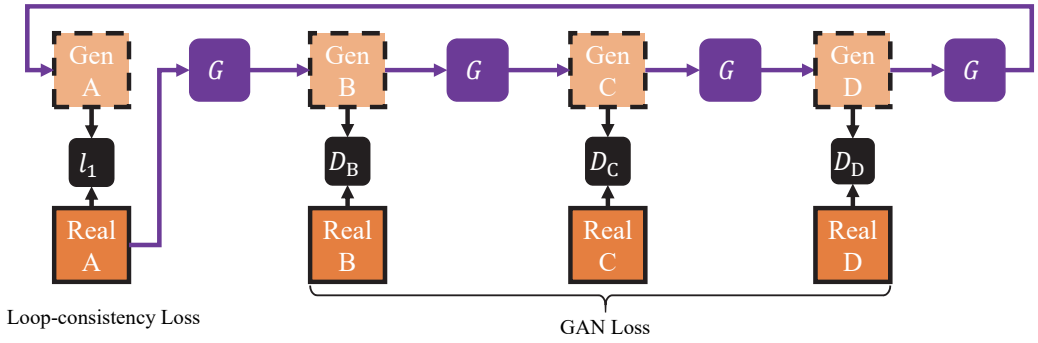

Figure 3.1: LoopGAN framework. For simplicity, only a single loop starting at one real domain in a four-domain problem is illustrated here. All four steps share a single generator G, parameterized by the step variable.

3.1.1 Adversarial Loss

Suppose $x_i \sim p_{data}(x_i)$. Then we expect that for all other domains j, $G^{||j-i||}(x_i, i)$ should be indistinguishable under D_j from (true) examples drawn from $p_{data}(x_j)$. Additionally, each D_j should aim to minimize the ability for G to generate examples that it cannot identify as fake. This forms the adversarial objective for a specific domain as:

$$\mathcal{L}_{GAN}(G, D_i) = \underset{x_i \sim p_{data}(x_i)}{E} [\log D_i(x_i)] + \sum_{j \neq i} \underset{x_j \sim p_{data}(x_j)}{E} \left[\log(1 - D_i(G^*(x_j)))\right]$$

where G^* denotes iteratively applying G until x_j is transformed into domain X_i, i.e. $||j-i||$ times. Taking this over all possible source domains, we get an overall adversarial objective as:

$$\mathcal{L}_{GAN}(G, D_1, \ldots, D_n) = \underset{i \sim q(i)}{E} \left[\underset{x_i \sim p_{data}(x_i)}{E} [\log D_i(x_i)] + \sum_{j \neq i} \underset{x_j \sim p_{data}(x_j)}{E} \left[\log(1 - D_i(G^*(x_j))) \right] \right]$$

where $q(i)$ is a prior on the set of domains, e.g., uniform.

3.1.2 Loop Consistency Loss

Within [ZPIE17], an adversarial loss was supplemented with a cycle consistency loss that ensured applying the generator from domain A to domain B followed by applying a *separate* generator from B to A acts like an identity function. However, LoopGAN only has a single generator and supports an arbitrary number of domains. Instead, we build a loop of computations by applying the generator G to a source image n times (equal to the number of domains being modeled). This constitutes loop consistency and allows us to reduce the set of possible transformations learned to those that adhere to the consistency condition. Loop consistency takes the form of an L_1 reconstruction objective for a domain X_i as:

$$\mathcal{L}_{Loop}(G, X_i) = \underset{x_i \sim p(x_i)}{E} ||x_i - G^n(x_i, i)||_1$$

3.1.3 Full Objective

The combined loss of LoopGAN over both adversarial and loop-consistency losses is becomes:

$$\mathrm{L(G,\, D}_1, \ldots, D_n, X_1, \ldots, X_n) = \mathcal{L}_{GAN}(G, D_1, \ldots, D_n) + \lambda E_{i \sim q(i)} [\mathcal{L}_{Loop}(G, X_i)]]$$
$$= E_{i \sim q(i)} \left[E_{x_i \sim p_{data}(x_i)} \left[\log D_i(x_i) \right] + \sum_{j \neq i} E_{x_j \sim p_{data}(x_j)} \left[\log \left(1 - D_i(G^*(x_j)) \right) \right] \right.$$
$$\left. + \lambda E_{x_i \sim p_{data}(x_i)} \|x_i - G^n(x_i)\|_1 \right]$$

where λ weighs the trade-off between adversarial and loop consistency losses.

An example instantiation of our framework for one loop in a four-domain problem is shown

in Figure 3.1. When training G, our objective function actually consists of four loops including $A \to B \to C \to D \to A$, $B \to C \to D \to A \to B$, $C \to D \to A \to B \to C$, and $D \to A \to B \to C \to D$. This is consistent with how CycleGAN is trained where two cycles are included.

3.2 Implementation

3.2.1 Network Architecture

We adopt the network architecture for style transfer proposed in [JAFF16] as our generator. This architecture has three main components: a down-sampling module $Enc(x)$, a sequence of residual blocks $T(h,i)$, and an up-sampling module $Dec(h)$. The generator G therefore is the composition $G(x,i) = Dec(T(Enc(x),i))$ where the dependence of T on i only relates to the step-specific AdaIN parameters [HB17] while all other parameters are independent of i. Following the notations from [JAFF16, ZPIE17], let c7-k denote a 7×7 Conv-ReLU layer with k filters and stride 1, dk denote a 3×3 Conv-ReLU layer with k filters and stride 2, Rk denote a residual block with two 3×3 Conv-AdaIn-ReLU layers with k filters each, uk denotes a 3×3 fractional-strided-Conv-LayerNorm-ReLU layer with k filters and stride $\frac{1}{2}$. The layer compositions of modules are down-sampling: c7-32, d64, d128; residual blocks: R128 $\times$ 6; up-sampling: u128, u64, c7-3. We use the PatchGAN discriminator architecture as [ZPIE17]: c4-64, c4-128, c4-256, c4-1, where c4-k denotes a 4×4 Conv-InstanceNorm-LeakyRelu(0.2) layer with k filters and stride 2.

3.2.2 Recurrent Transformation

Suppose we wish to translate some $x_i \in X_i$ to another domain X_j. A naive approach would formulate this as repeated application of G, $|j-i|$ times. However, referencing our definition of G, we can unroll this to find that we must apply Enc and Dec $j-i$ times throughout the computation. However, Enc and Dec are only responsible for bringing an observation into and out of the space

of T. This is not only a waste of computation when we only require an output at X_j, but it has serious implications for the ability of gradients to propagate through the computation. Therefore, we implement $G(x_i, i)$ as: a single application of $Enc(x_i)$, $j-i$ applications of $T(h)$, and a single application of $Dec(h)$. T is applied *recurrently* and the entire generator is of the form:

$$G(x_i, i) = Dec(T^{|j-i|}(Enc(x_i)))$$

We show in our ablation studies that this re-formulation is critical to the learning process and the resulting quality of the transformations learned. Additionally, $T(h, i)$ is given a a set of separate, learnable normalization (AdaIN [HB17]) parameters that it selects based off of i with all other parameters of T being stationary across time steps. The overall architecture is shown in Figure 3.2.

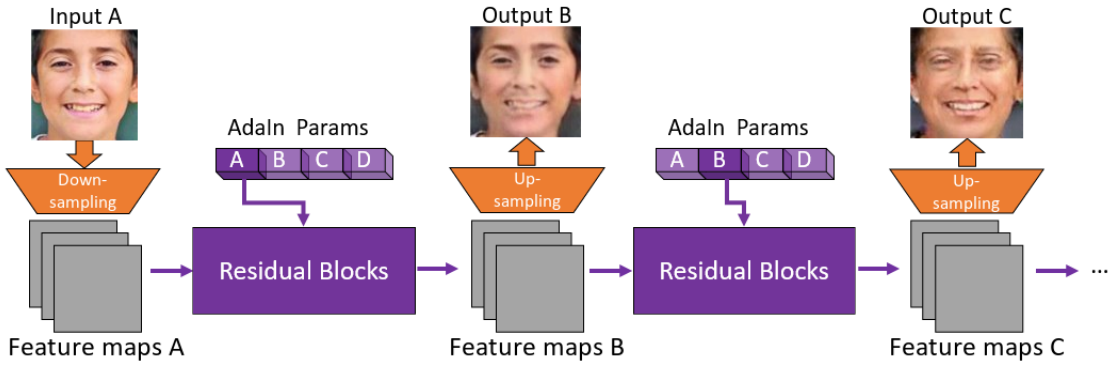

Figure 3.2: LoopGAN network. All modules share parameters.

3.2.3 Training

For all datasets, the loop-consistency loss coefficient λ is set to 10. We use Adam optimizer [KB14] with initial learning rate of 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$. We train the face aging dataset and Alps seasons dataset for 50 epochs and 70 epochs respectively with initial learning rate and linearly decay learning rate to 0 for 10 epochs for both datasets.

3.3 Experimental Results

We apply LoopGAN to two very different sequential image generation tasks: face aging and chaging seasons of scenery pictures. Baselines are built with two bi-domain models, CycleGAN[ZPIE17] and UNIT[LBK17] and also a general-purpose multi-domain model StarGAN[CCK$^+$18]. We are interested in the sequential transformation capabilities of separately trained bi-domains compared to LoopGAN. Therefore, for each of the two bi-domains models, we train a separate model between every pair of sequential domains, i.e. X_i and X_{i+1} and additionally train a model between every pair (not necessarily sequential) domains X_i and X_j ($i \neq j$). The first approach allows us to build a baseline for sequential generation by *chaining* the (separately learned) models in the necessary order. For instance, if we have four domains: A, B, C, D, then we can train four separate CycleGAN (or UNIT) models: $G_{AB}, G_{BC}, G_{CD}, G_{DA}$ and correctly compose them to replicate the desired sequential transformation. Additionally, we can train direct versions e.g. G_{AC} of CycleGAN (or UNIT) for a more complete comparison against LoopGAN. We refer to composed versions of separately trained models as *Chained-CycleGAN* and *Chained-UNIT* depending on the base translation method used. Since StarGAN[CCK$^+$18] inherently allows transformation between any two domains, we can apply this in a chained or direct manner without any additional models needing to be trained.

3.3.1 Face Aging

We adopt the UTKFace dataset [ZQ17] for modeling the face aging task. It consists of over 20,000 face-only images of different ages. We divide the dataset into four groups in order of increasing age according to the ground truth age given in the dataset as A consisting of ages from 10-20, B containing ages25-35, C containing ages 40-50, and D containing ages 50-100. The number of images for each group are 1531, 5000, 2245, 4957, respectively, where a 95/5 train/test split is made. The results of LoopGAN generation are shown in on the left side in Figure 3.3.

LoopGAN shows advantage over baseline models in two aspects. The overall facial

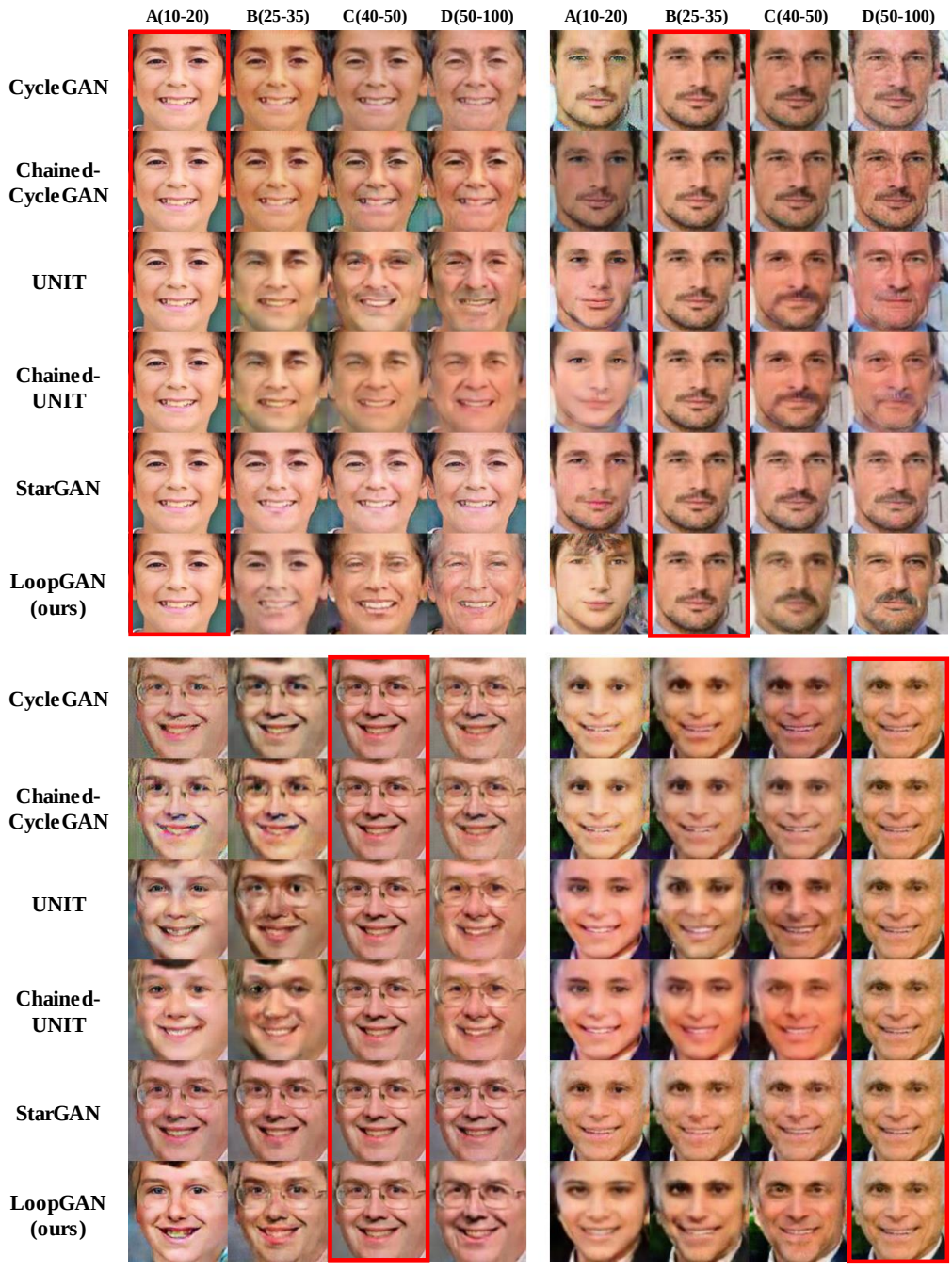

Figure 3.3: Face Aging with LoopGAN compared to baselines. Input real images are highlighted with rectangles.

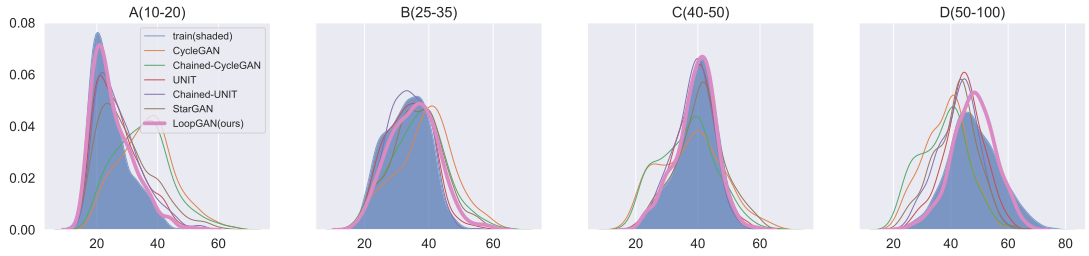

Figure 3.4: Comparing estimated age distribution between model generated images and train images.

structure is preserved better than other models which we believe is due to the enforced loop consistency loss. Moreover, LoopGAN is able to make more apparent age changes compared to the rest of baseline models. Baseline models struggle in either of the two aspects. CycleGAN models preserve facial structure well due to the strong cycle consistency constraint, however they struggle to make apparent age changes for the same reason. UNIT models on the other hand only implicitly enforce cycle consistency, thus are able to make much more apparent age changes. But this flexibility comes at the cost of less facial structure preserved from the input image.

In order to quantitatively compare the amount of age change between models, we obtain an age distribution of *generated* images by running a pre-trained age estimator DEX [RTVG15]. The estimated age distributions of generated images (from input test images) are compared against those of the train images in Figure 3.4. The age distribution of LoopGAN generated images is closer to that of the train images across all four age groups when compared to the baseline models — suggesting that it more faithfully learns the sequential age distribution changes of train data.

3.3.2 Changing Seasons

We use the collected scenery photos of Alps mountains of four seasons from [AATVG18]. They are ordered into a sequence starting from Spring (A), to Summer (B), Fall (C), and Winter (D). They each have approximately 1700 images and are divided into 95/5% training and test set.

We show the results in Figure 3.5. Overall, LoopGAN is able to make drastic season change while maintaining the overall structure of the input scenery images.

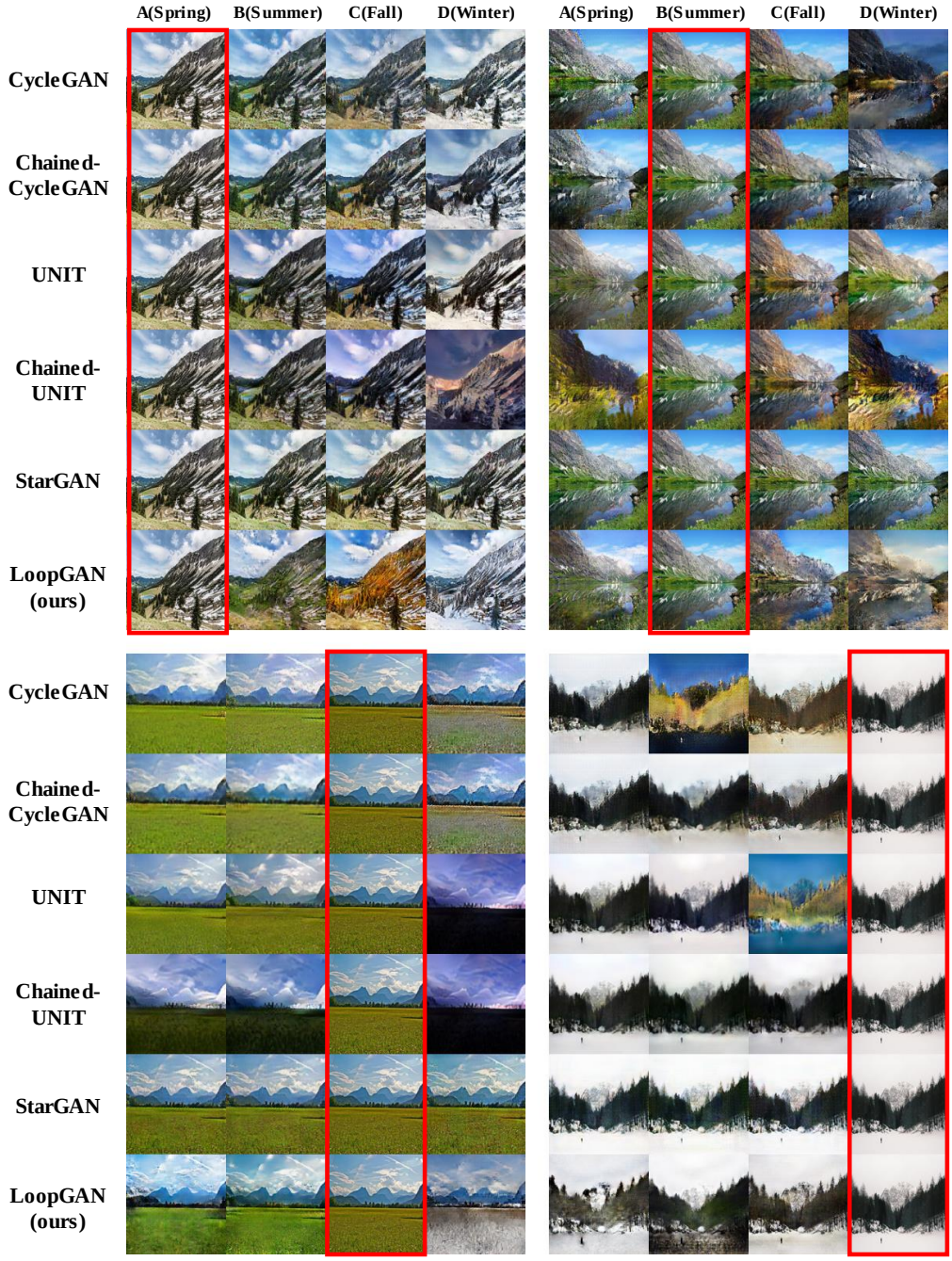

Figure 3.5: Alps season changes with LoopGAN compared to baselines. Input real images are highlighted with rectangles.

3.4 Model Features

3.4.1 Stable Loop Dynamics

LoopGAN models sequential visual phenomena by assuming consistent periodic dynamics across the stages of the sequence. How well does LoopGAN learn such dynamics? And do chained cycle-consistent bi-domain models such as Chained-CycleGAN end up learning a similar dynamic without the need for explicit loop consistency? To understand these questions, we iteratively apply LoopGAN and Chained-CycleGAN in a loop 20 times. This process uses the last generated image from previous loop to generate a new loop. The results are shown in Figure 3.6. In loops 1 and 2, both models are generating high-quality images that are consistent across loops. At loop 5, Chained-CycleGAN images are getting blurry while LoopGAN images are the same as loop 1 and 2. At loop 20, Chained-CycleGAN images are unrecognizable while LoopGAN images only suffer minor resolution decrease. This suggests that LoopGAN indeed learns a stable periodic dynamic. Chained bi-domain models with only cycle-consistency also demonstrate a form of periodic dynamics albeit one that becomes unstable substantially faster than LoopGAN. Chained approaches appear to suffer from exposure bias issues as they are exposed to *generated* images after the first step instead of ground-truth inputs (as during training) whereas LoopGAN directly tackles this issue in order to achieve loop consistency.

3.4.2 Architectures

We experiment with several network architecture variations and investigate their effect on generation quality. First, attention mechanisms have proven to be useful in GAN image generation [ZGMO18]. In the context of tasks involving sequential images, [WGGH18] showed that applying attention mechanisms in both the space and time dimensions helps increase video classification accuracy. Combining these two results, we added an attention module to the residual blocks that attend to not only the features maps of the same layer (space) but also the same layer

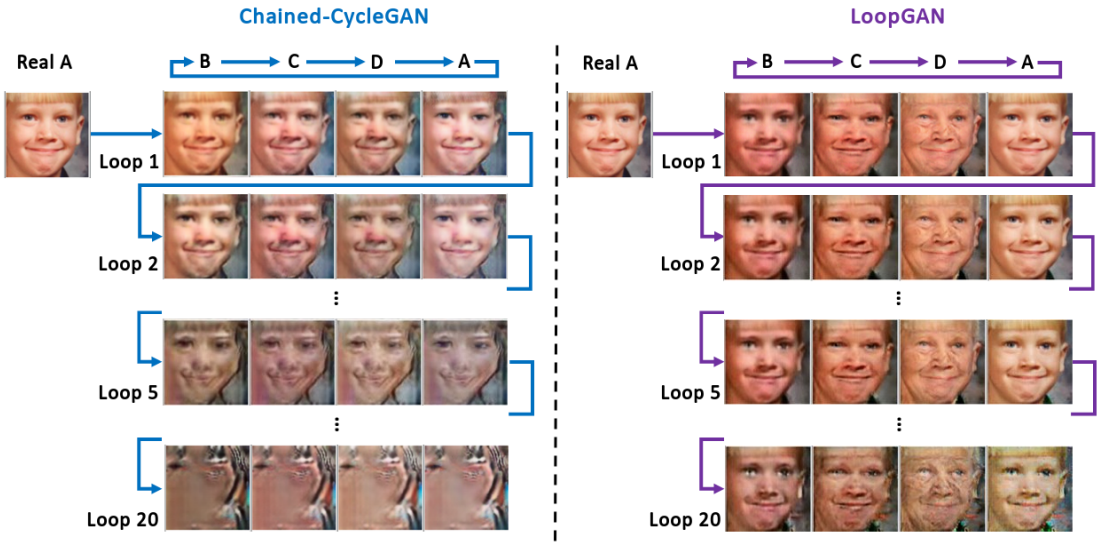

Figure 3.6: Learned models applied recursively to an input real image. The left and right side show images generated by Chained-CycleGAN and LoopGAN respectively. LoopGAN learns a stable loop dynamic compared to Chained-CycleGAN.

of all previous generation steps (time). However, we found that the network struggles to generate high quality image after adding this type of attention mechanism. We also noticed that [HLBK18] used a similar network architecture but with different normalization layers. The authors mentioned that for down-sampling, it is better to use no normalization to preserve information from input image, and for up-sampling it is better to use layer-normalization for faster training and higher quality. We applied these changes and found that they indeed help the network produce better results. The results under these variations are shown in Figure 3.7 (first three rows).

Table 3.1: Model size comparison. * Note that the parameter count for vanilla and chained versions of bi-domain models (CycleGAN, Chained-CycleGAN, UNIT, and Chained-UNIT) are totals of separate pair-wise generators that together facilitate sequence generation.

Model	**Parameter Count**
CycleGAN	94.056 M *
Chained-CycleGAN	62.704 M *
UNIT	133.680 M *
Chained-UNIT	89.120 M *
StarGAN	8.427 M
LoopGAN(ours)	11.008 M

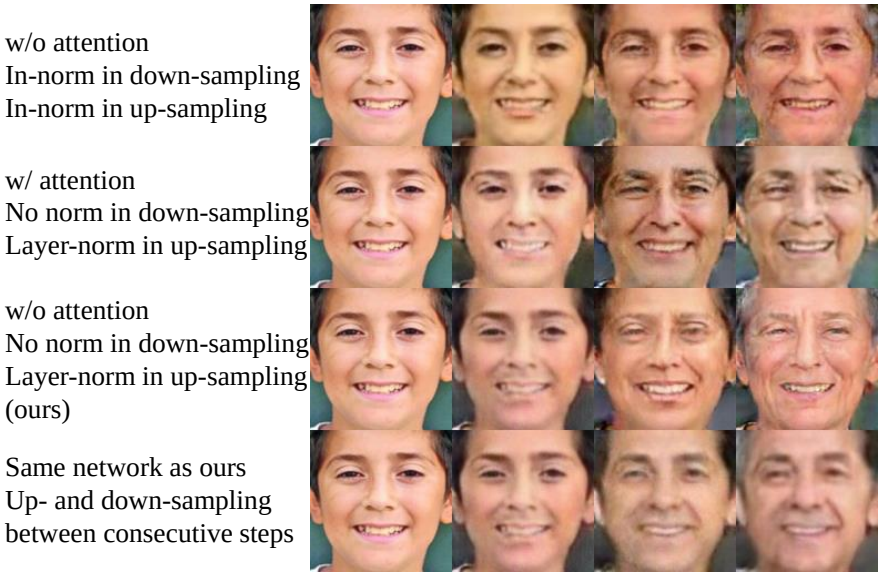

Figure 3.7: Ablation study for the architecture changes.

Moreover, we show the the importance of the recurrent form of $T(h)$ discussed in Section 3.2.2. We compare the choice to invoke *Enc* and *Dec* at each time step versus applying them once with some number of recurrent applications of T in Figure 3.7 (last row) and show the poor quality observed when performing the loop naively.

Lastly, we calculate the parameter count of generator networks compared in the face aging and season change experiments above and show that our final generator network architecture is parameter-efficient compared to baseline models in Table 3.1. This parameter-efficiency is mainly due to the fact that every transformation step uses a shared network as opposed to its own separate network. Less parameters also act as an added layer of regularization and accelerate training.

For completeness, a selection of failure cases on face aging task are shown in Figure 3.8 and Figure 3.9. Input images and failure generations are highlighted respectively in red and purple (viewed in color). In both cases, the highlighted generated images (the first column in (a) and the last column in (b)) bear some semantic dissimilarity to the input images. It seems that sometimes the network overfit to more drastic transformations that only preserve overall facial structure and orientation but neglects all other features.

Figure 3.8: Failure case 1.

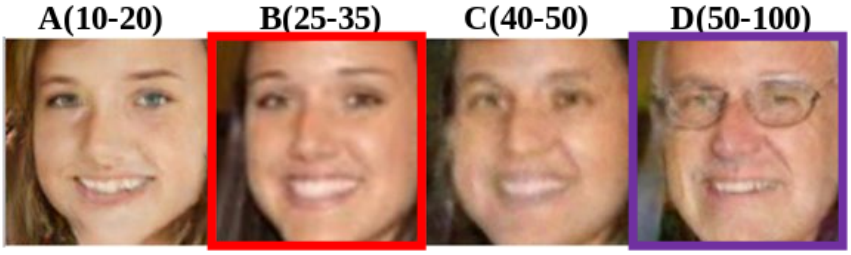

Figure 3.9: Failure case 2.

3.4.3 Linear Interpolation

The AdaIn parameters [HB17] achieve affine transformations in each layer with two channel-wise parameters γ, β, where the output of AdaIn layers can be expressed as,

$$y_{AdaIn}(x, \gamma, \beta) = \gamma \frac{x - mean(x)}{std(x)} + \beta.$$

We hypothesize that by weighting these two parameters by $\sigma \in [0, 1]$ at test time can possibly achieve linear interpolation between input and the full transformation ($\sigma = 1$). To see the effect of this interpolation scheme, we generate an extra interpolated transformation between every two transformation stages by weighting the AdaIn parameters with $sigma = 0.5$.

The results on the Alps changing seasons dataset are shown in Figure 3.10. The input (spring images) are highlighted with red rectangles and the full transformations (summer, fall, and winter) are highlighted with purple rectangles. The images in between are interpolated transformation obtained by weighting the AdaIn parameters with $sigma = 0.5$. The results show that this simple linear weighting can generate interpolation at test time with good quality. Note

Figure 3.10: Linear interpolation on AdaIn parameters between transformation stages.

that this process is done completely at test time, suggesting that the AdaIn parameters somehow learned a continuous representation of the sequential transformation albeit with discrete data.

This feature enhances our model's application scope and shows promise in applying our model to areas such as video prediction and video synthesis.

3.4.4 Generating Multiple Sequences

Multi-modal transformation is a central problem in image-to-image transformation. There often exists more than one feasible transformation from one image domain to another. So if the transformation function is defined to be one-to-one, then no matter how the training process is carried out, it can not learn a good transformation as it may be confused by parallel signals. For example, the same young person's face can be transformed into old with or without glasses and the model should at least have the capacity to learn this. This has largely been solved in image-to-image transformation with models proposed in [HLBK18] and [ARS+18]. The key

idea is to condition the transformation function on not only the input image but also a random vector drawn from a pre-determined distribution.

Sequential image transformation also requires multi-modal. This is not hard to see if we take the same face aging example. As the model transforms a young person's face through multiple stages, adding a pair of glasses at any point is semantically meaningful. Thus, we also explore architecture changes that enable the model to do this. We look at architecture proposed in [KLA19] where the AdaIn parameters are functions of an input noise vector. Since the AdaIn parameters control transformation, this setup allows transformation to be conditioned on a noise vector. The architecture is shown in Figure 3.11. Note that noise vector z controls all steps of transformation in the sequence and not just a single step.

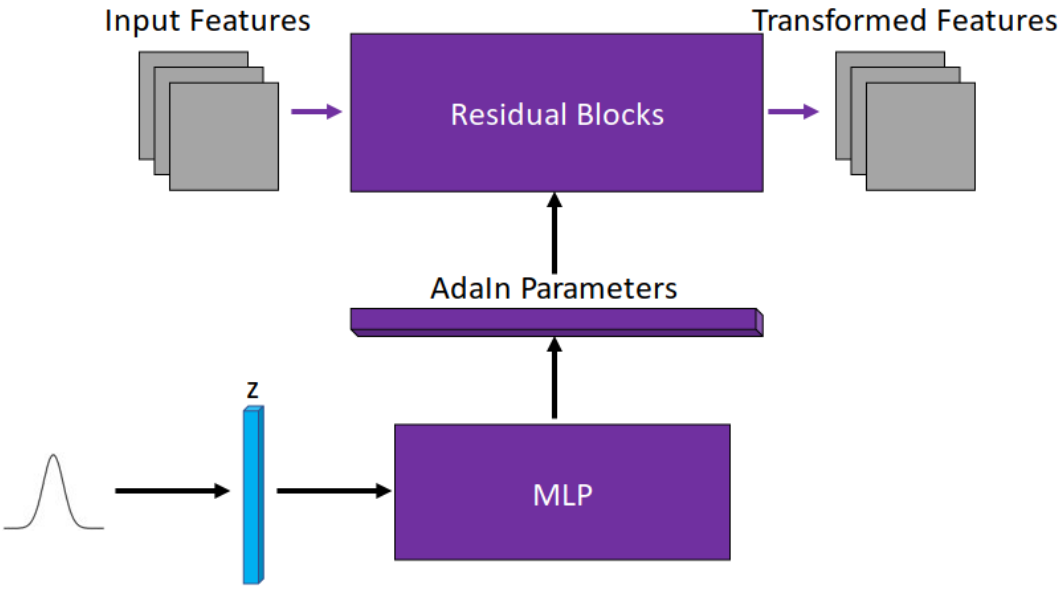

Figure 3.11: Sequence transformation conditioned on a noise vector.

However, experiments show that this architecture does not produce a variety of sequences as expected. We believe that this is due to lack of hyper parameter tuning which requires future work.

Chapter 4

Conclusion

This thesis presents an in-depth study on sequential image transformation without aligned data. This is a problem yet to be studied in-depth and has a lot of applications. The related works on GAN-based unpaired image-to-image transformation models are introduced. The shortcomings of these existing image-to-image models on sequential image transformation are discussed and showcased through experiments.

We proposed an extension to the family of image-to-image translation methods that directly model sequential image transformation. We require that the translation task can be modeled as a consistent loop. This allows us to use a shared generator across all time steps leading to significant efficiency gains over a nave chaining of image-to-image transformation models. Despite this, our architecture compares favorably and even shows stable dynamics of the both the face aging and seasonal phenomena modeled, when compared with the classic image-to-image transformation models such CycleGAN and UNIT.

I would like to acknowledge Professor Zhuowen Tu for his support as the chair of my committee. Through multiple drafts and many long nights, his guidance has proved to be invaluable.

This entire thesis, in part is currently being prepared for submission for publication of the

material. Siyang Wang; Justin Lazarow; Kwonjoon Lee; Zhuowen Tu. The thesis author was the primary investiator and author of this material. Thanks to my co-authors.

I would also like to thank Weijian Xu for giving me countless helpful discussions.

Bibliography

[AATVG18] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018.

[ARS+18] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *arXiv preprint arXiv:1802.10151*, 2018.

[CCK+18] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

[DSK17] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2, 2017.

[GEB15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[GJFF+18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[HLBK18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KCK+17] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[LBB+98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBK17] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[LTH+18] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018.

[PATVG18] Sveinn Palsson, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Generative adversarial style transfer networks for face aging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2084–2092, 2018.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[RTVG15] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015.

[SCW+15] Xingjian Shi, 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*, pages 802–810, 2015.

[SMS15] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[VPT16] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[VT17] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1020–1028, 2017.

[WGGH18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[WJY+19] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *ICLR*, 2019.

[ZGMO18] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[Zhu19] Jun-Yan Zhu. Learning to synthesize images. UCSD CSE Colloquium Lecture Talk (April 22nd, 2019), 2019.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[ZQ17] Song Yang Zhang, Zhifei and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.