Controllable Visual Synthesis

Badour AlBahar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Jia-Bin Huang, Co-chair
A. Lynn Abbott, Co-chair
Harpreet S. Dhillon
Ruoxi Jia
Ismini Lourentzou
Ting-Chun Wang

March 30, 2023
Blacksburg, Virginia

Keywords: Computer vision, Visual synthesis, Image generation

Controllable Visual Synthesis

Badour AlBahar

ABSTRACT

Computer graphics has become an integral part of various industries such as entertainment (i.e.,films and content creation), fashion (i.e.,virtual try-on), and video games. Computer graphics has evolved tremendously over the past years. It has shown remarkable image generation improvement from low-quality, pixelated images with limited details to highly realistic images with fine details that can often be mistaken for real images. However, the traditional pipeline of rendering an image in computer graphics is complex and time-consuming. The whole process of creating the geometry, material, and textures requires not only time but also significant expertise. In this work, we aim to replace this complex traditional computer graphics pipeline with a simple machine learning model. This machine learning model can synthesize realistic images without requiring expertise or significant time and effort. Specifically, we address the problem of controllable image synthesis. We propose several approaches that allow the user to synthesize realistic content and manipulate images to achieve their desired goals with ease and flexibility.

Controllable Visual Synthesis

Badour AlBahar

GENERAL AUDIENCE ABSTRACT

Computer graphics has become an integral part of various industries such as entertainment (i.e.,films and content creation), fashion (i.e.,virtual try-on), and video games. Computer graphics has evolved tremendously over the past years. It has shown remarkable image generation improvement from low-quality, pixelated images with limited details to highly realistic images with fine details that can often be mistaken for real images. However, the traditional process of generating an image in computer graphics is complex and time-consuming. You need to set up a camera and light, and create objects with all sorts of details. This requires not only time but also significant expertise. In this work, we aim to replace this complex traditional computer graphics pipeline with a simple machine learning model. This machine learning model can generate realistic images without requiring expertise or significant time and effort. Specifically, we address the problem of controllable image synthesis. We propose several approaches that allow the user to synthesize realistic content and manipulate images to achieve their desired goals with ease and flexibility.

Dedicated to my children.

Acknowledgments

All praise is due to Allah, by whose favor good deeds are accomplished.

I would like to thank my advisor, Jia-Bin Huang, for his guidance and support. I appreciate all the time and effort he contributed to this journey. Thank you for introducing me to this fascinating realm of research.

I would like to extend my gratitude to my lab colleagues for their valuable insights and discussions. I am grateful for their support and collaboration.

I would also like to thank Kuwait University for choosing me among many highly qualified individuals for their generous scholarship.

Finally, I would like to express my heartfelt appreciation to my family for their love and endless support and encouragement. To my parents, thank you for being continuously supportive of me and my goals in life. To my brothers and sisters, thank you for believing in me and for being there when I need you most. To my husband and children, thank you for your love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Computer graphics has become an essential tool for various industries such as entertainment (i.e.,films and content creation), fashion (i.e.,virtual try-on), and video games. Over the years, computer graphics has evolved tremendously. Initially, computer graphics produced images that were of low-quality and pixelated with very little details. However, with technological advancements, computer graphics systems are now able to create high-quality images and videos with fine details that can be easily mistaken for real images.

The traditional process of rendering an image in computer graphics involves a series of steps that can be quite complex and time-consuming. This process involves setting up a virtual camera and light source, and creating a detailed object representation with geometry, material, textures, and shading, which requires not only time but also significant skill and expertise.

In this work, we aim to replace this complex traditional computer graphics pipeline with a simple machine learning model. This machine learning model can synthesize realistic images without requiring expertise or significant time and effort. Specifically, we address the problem of controllable image synthesis. In such problems, the user provides an image and specifies a control signal to guide the synthesis of realistic images. These control signals

can specify the pose in pose-conditional human synthesis, a texture patch in sketch texture synthesis, a high-resolution image in depth upsampling, or a novel view for human view synthesis. We propose several approaches that allow the user to synthesize realistic content and manipulate images to achieve their desired goals with ease and flexibility. Our proposed approaches help speed up the process of image generation and can make it easier for people without a background in computer graphics to produce high-quality images for their specific tasks.

We study the following image synthesis problems:

Controllable synthesis. We tackle controllable image-to-image translation problems. In such problems, we aim to translate an input image into a corresponding output image while respecting the constraints specified in the provided guidance image. This guidance can take various different forms (e.g.,color strokes, sketch, texture patch, image, and mask). We have presented a new conditional scheme for guided image-to-image translation problems. Our core technical contributions lie in the use of *spatially varying* feature transformation and the design of *bi-directional conditioning* scheme that allow the mutual modulation of the guidance and input network branches. While being application-agnostic, our approach achieves competitive performance with the state-of-the-art. The generality of our method opens promising direction of incorporating a wide variety of constraints for image-to-image translation problems.

High-resolution synthesis. We tackle high-resolution pose-guided person image generation. We present a simple yet effective approach for single-image human reposing guided by arbitrary body shapes and poses. Our core technical novelties lie in 1) spatial modulation of a pose-conditioned StyleGAN generator and 2) a symmetry-guided inpainting network for completing correspondence field. We demonstrate that our approach is capable of syn-

thesizing *photo-realistic* images in the desired target pose and *preserving details* from the source image. We validate various design choices through an ablation study and show improved results when compared with the state-of-the-art human reposing algorithms. Our controllable human image synthesis approach enables high-quality human pose transfer and garment transfer, providing a promising direction for rendering human images.

Time-consistent synthesis. We tackle temporally consistent editing of the semantic attributes of a given human in a video. We utilize the impressive image semantic editing capability of Generative Adversarial Networks (GANs). Applying these GAN-based editing to a video independently for each frame inevitably results in temporal flickering artifacts. We present a simple yet effective method to facilitate temporally coherent video editing. Our core idea is to minimize the temporal photometric inconsistency by optimizing both the latent code and the pre-trained generator. We show that the resulting edited video maintains the identity of the human and the desired edit direction while still preserving the temporal consistency. Furthermore, our method is model-agnostic, allowing for its application with different GAN inversion and manipulation techniques.

1.2 Organization

In Chapter 2, we address the problem of guided image-to-image translation problems where we translate an input image into another while respecting the constraints provided by an external, userprovided guidance image. These controllable image-to-image translation problems often require task-specific architectures and training objective functions as the guidance can take various different forms (e.g.,color strokes, sketch, texture patch, image, and mask). We introduce a new conditioning scheme for controlling image synthesis using available guidance signals and demonstrate applicability to several sample applications, including person

image synthesis guided by a given pose, sketch-to-photo synthesis guided with a texture patch, and depth upsampling guided with an RGB image. The proposed conditioning scheme uses ResNet or UNet architectures as the model's base architecture. These simple convolutional neural network (CNN) decoders have limitations in synthesizing high-resolution detailed content.

Therefore, in Chapter 3, we focus on synthesizing high-resolution content. We present an algorithm for re-rendering a person from a single image under arbitrary poses. We achieve this by first learning to inpaint the correspondence field between the body surface texture and the source image with a human body symmetry prior. The inpainted correspondence field allows us to transfer/warp local features extracted from the source to the target view even under large pose changes. Directly mapping the warped local features to an RGB image using a simple CNN decoder often leads to visible artifacts. Thus, we extend the StyleGAN generator so that it takes pose as input (for controlling poses) and introduce a spatially varying modulation for the latent space using the warped local features (for controlling appearances). Although we are capable of realistically hallucinating occluded content while preserving the identity and fine details of the source image, the resulting synthesized image is limited to 2D space and does not support temporally consistent synthesis nor view synthesis.

Therefore, in Chapter 4, we focus on temporally consistent editing of a human from a given video. To address this problem, we leverage image-based GAN inversion and editing capability of real images, e.g., changing object classes, modifying attributes, or transferring styles. Using these GAN-based editing techniques on each frame of a video independently can lead to temporal flickering artifacts. We refine the flickering results using a flow-based method to minimize the bi-directional photometric loss by optimizing both the latent code and the pre-trained generator.

Finally, in Chapter 5, we present the conclusions and outline potential directions for future

research.

Chapter 2

Guided Image-to-Image Translation with Bi-Directional Feature Transformation

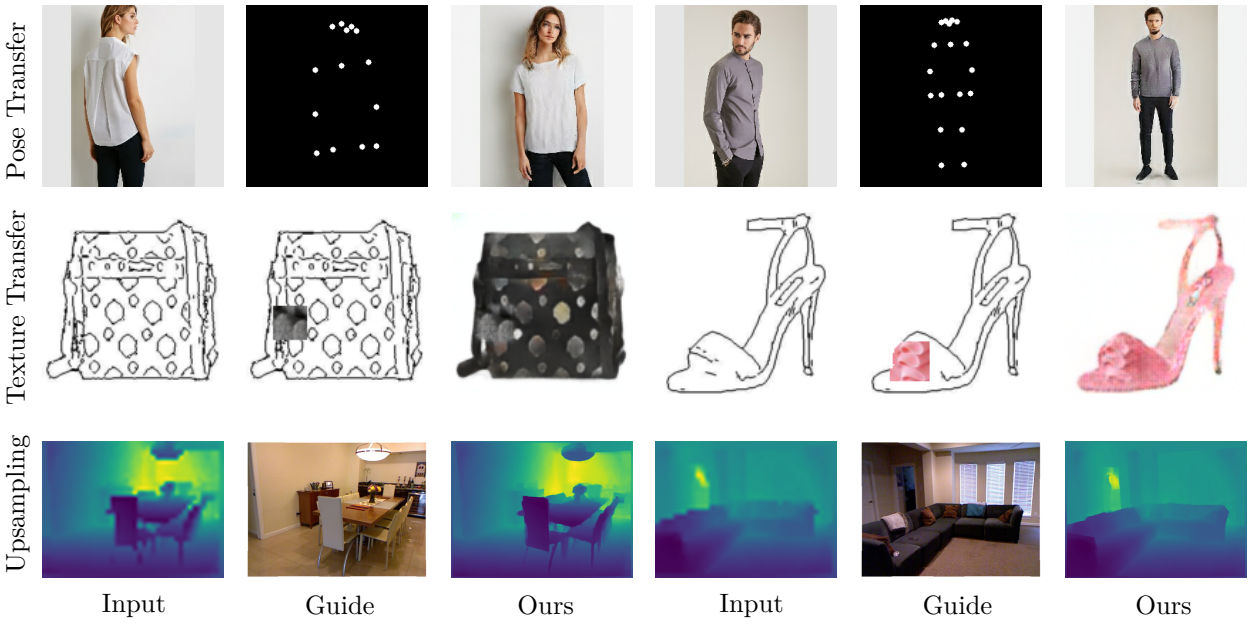


Figure 2.1: **Applications of guided image-to-image translation.** We present an algorithm that translates an input image into a corresponding output image while respecting the constraints specified in the provided guidance image. We introduce a new conditioning scheme for controlling image synthesis using available guidance signals and demonstrate applicability to several sample applications, including person image synthesis guided by a given pose (*top*), sketch-to-photo synthesis guided with a texture patch (*middle*), and depth upsampling guided with an RGB image (*bottom*).

We address the problem of guided image-to-image translation where we translate an input image into another while respecting the constraints provided by an external, user-provided guidance image. Various conditioning methods for leveraging the given guidance image have been explored, including input/feature concatenation and conditional affine transformation of feature activations. All these conditioning mechanisms, however, are uni-directional, i.e., no information flow from the input image back to the guidance. To better utilize the constraints of the guidance image, we present a bi-directional feature transformation (bFT) scheme. We show that our bFT scheme outperforms other conditioning schemes and has comparable results to state-of-the-art methods on different tasks.

2.1 Introduction

In an image-to-image translation problem [21], we aim to translate an image from one domain to another. Many problems in computer vision, graphics, and image processing can be formulated as image-to-image translation tasks, including semantic image synthesis, style transfer, colorization, sketch to photos, to name a few. An extension to these image-to-image translation problems involves an additional *guidance image* that helps achieve controllable translation. A guidance image typically reflects the desired visual effects or constraints specified by a user or provides additional information via other modalities (color/depth, flash/non-flash, color/IR). A guidance image can thus take many different forms, e.g. color strokes or palette, semantic labels, texture patch, image, or mask. As such, most of the existing solutions for such problems often have application-specific architectures and objective functions, and consequently cannot be directly applied to other problems.

The main technical question for guided image-to-image translation problems is how the conditional guidance image is used to affect the processing of the input source image. Various

forms of conditioning schemes have been proposed in the literature. The most common one is to directly concatenate the input source image and the guidance image at the input level (i.e., concatenation along the channel dimension). While being parameter efficient, this approach assumes that the additional guidance is required at the input level and the information can be carried through all the subsequent layers. Another commonly used alternative is to concatenate the guidance and the input information at the feature level, assuming that the guidance feature representation is required at a certain level within the model.

A recent generalized conditioning scheme formalized as Feature-wise Linear Modulation (FiLM) has been successfully applied in visual reasoning task [22]. In this scheme, affine transformations are applied to intermediate feature activations using scaling and shifting parameters learned from some external conditional information. In this approach, the learned scaling and shifting operations are applied *feature-wise* (i.e., spatially invariant). There are other conditioning approaches similar to FiLM that have shown effectiveness in the context of style transfer. In this task, given an input image and a guidance style image, the goal is to synthesize an image that combines the content of the input image with the style of the guidance image. One such approach is conditional instance normalization (CIN) [23], which can be seen as a FiLM layer replacing a normalization layer. In CIN, the feature representation is first normalized to zero mean and unit standard deviation. Then an affine transformation is applied to the normalized feature representation using scaling and shifting parameters learned from the guidance style image. Another approach is adaptive instance normalization (AdaIN) [24]. AdaIN is very similar to CIN, however, unlike CIN, it does not learn the affine transformation parameters but uses the mean and standard deviation of the guidance style image as the scaling and shifting parameters respectively.

In this work, we propose a generalized conditioning scheme to incorporate the guidance image into the image-to-image translation model and show its applicability to different ap-

plications. There are two key differences between our proposed approach and the existing conditioning schemes. First, we propose to apply the conditioning operation in *both* direction with information flowing not only from the guidance image to the input image, but from the input image to the guidance image as well. Second, we extend the existing feature-wise feature transformation to be *spatially varying* to adapt to different contents in the input image. We refer to our proposed approach as bi-directional feature transformation (bFT). We validate the design of bFT through extensive experiments across multiple applications, including pose guidance appearance transfer, image synthesis with texture patch guidance, and joint depth upsampling. We demonstrate that our method, while not application-specific, achieves competitive or better performance than the state-of-the-art. Through extensive ablation study, we also show that the proposed bFT is more effective than commonly used conditional schemes such as input/feature concatenation, CIN [23] and AdaIN [24].

We make the following two contributions. First, we present the *bi-directional* feature transformation for generic guided image-to-image translation tasks. Compared to existing approaches that only allow the information flow from guidance to the source image, we show that incorporating the information from the input to the guidance further help improve the performance of the end task. Second, we propose a *spatially varying* extension of feature-wise transformation to better capture local contents from the guidance and the source image.

2.2 Related Work

Image-to-image translation A generative model is an approach to learn a data distribution to generate new samples. One widely used technique is generative adversarial networks (GANs) [25]. In GANs, there is a generator that tries to generate samples that look realistic to fool the discriminator, which tries to accurately tell whether a sample is real or fake.

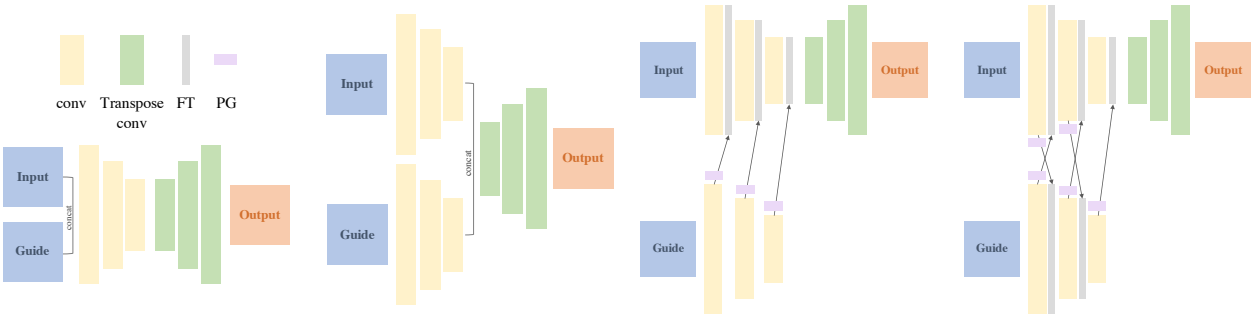
Conditional GANs extend the GANs by incorporating conditional information. One specific application of conditional GANs is image-to-image translation [21, 26, 27]. Several recent advances include learning from unpaired dataset [28, 29, 30], improving diversity [31, 32, 33], application to domain adaptation [34, 35, 36], and extension to video [37].

Our work builds upon the recent advances in image-to-image translation and aims to extend it to a broader set of controllable image synthesis problems. We develop our network architecture similar to that of the pix2pix [21], but the proposed bi-directional and spatially varying feature transformation layer is network-agnostic.

Guided image-to-image translation A variant of image-to-image translation problem is to incorporate additional guidance image. In a guided image-to-image translation problem, we aim to translate an image from one domain into another while respecting certain constraints specified by a guidance image. This guidance image can take many forms. Examples include color strokes [38, 39], patches [40], or color palette [41] to aid in user-guided colorization. The guidance can also be a domain label, as in a multi-domain image-to-image translation [42]. Another form could be a style image as in the problem of style transfer [23, 24, 43], a texture patch to texturize a sketch image [1], or a high-resolution RGB image to aid in depth upsampling [44, 45]. Moreover, the guidance signal could be the multi-channel and sparse, such as pose landmark for pose guided person image synthesis problems [46, 47, 48, 49]. The guidance could also be a mask and sketch enabling users to inpaint and manipulate images [50]. Due to the many different possible forms of the guidance images, most of the existing solutions for this class of problems are tailored toward specific applications, e.g., with specifically designed network architectures and training objectives.

Compared to many existing efforts in guided image-to-image translation, we focus on developing a conditioning scheme that is *application-independent*. This makes our technique

more widely applicable to many tasks with different forms of guidance.



(a) Input Concatenation (b) Feature Concatenation (c) Uni-directional FT (d) Ours: Bi-directional FT

Figure 2.2: **Conditioning schemes.** There are many schemes to incorporate the additional guidance into the image-to-image translation model. One straight forward scheme is (a) input concatenation, this will assume that we need the guidance image at the first stage of the model. Another scheme is (b) feature concatenation. It assumes that we need the feature representation of the guide before upsampling. In (c) we replace every normalization layer with our novel feature transformation (FT) layer that manipulates the input using scaling and shifting parameters generated from the guide using a parameter generator (PG). We denote this uni-directional scheme as uFT. In this work, we propose (d) a bi-directional feature transformation scheme denoted as bFT. In bFT, the input is manipulated using scaling and shifting parameters generated from the guide and the guide is also manipulated using scaling and shifting parameters generated from the input.

Conditioning schemes Figure 2.2 compares with several commonly used conditioning schemes. The most straightforward way of performing guided image-to-image translation is to concatenate the input and the guidance image (along the feature channel dimension), followed by conventional image-to-image translation models. Such an input concatenation approach can be viewed as a simple conditioning scheme. This approach assumes that the guidance signals are required from the input stage [1, 40, 50]. Several other types of conditioning schemes have been proposed in the literature. Instead of concatenating the guidance and the input image at the input, one can also concatenate their feature activations at a certain layer [45, 51]. However, it may be non-trivial to choose a suitable level of the layer to concentrate input/guidance features for subsequent processing. A recent and a more general

scheme, Feature-wise Linear Modulation (FiLM) [22], applies feature-wise affine transformation using scaling and shifting parameters generated from conditioning information. Such a scheme has shown improved performance when applied to the problem of visual reasoning. Other variations of FiLM have shown good performance in the context of style transfer. Those approaches can be seen as replacing a normalization layer with a FiLM layer. One notable approach is the conditional instance normalization (CIN), where the scaling and shifting parameters are learned [23]. Another approach is adaptive instance normalization (AdaIN) where instead of learning the scaling and shifting parameters, the mean and standard deviation from the guidance features are used directly [24].

Unlike existing conditioning schemes that allow information flow only from the guidance to the input (i.e., uni-directional conditioning), we show that the proposed *bi-directional conditioning* method leads to sizable performance improvement. Furthermore, we generalize the existing spatially invariant feature-wise transform methods to support *spatially varying* transformation.

2.3 Bi-Directional Feature Transformation

In this work, we aim to translate an image from one domain to another while respecting the constraints specified by a given guidance image. To tackle this problem, we propose Bi-Directional Feature Transformation (bFT) to incorporate the additional guidance image into the conditional generative model. We show that this conditioning scheme can be applied to various guided image-to-image translation problems without application-specific designs.

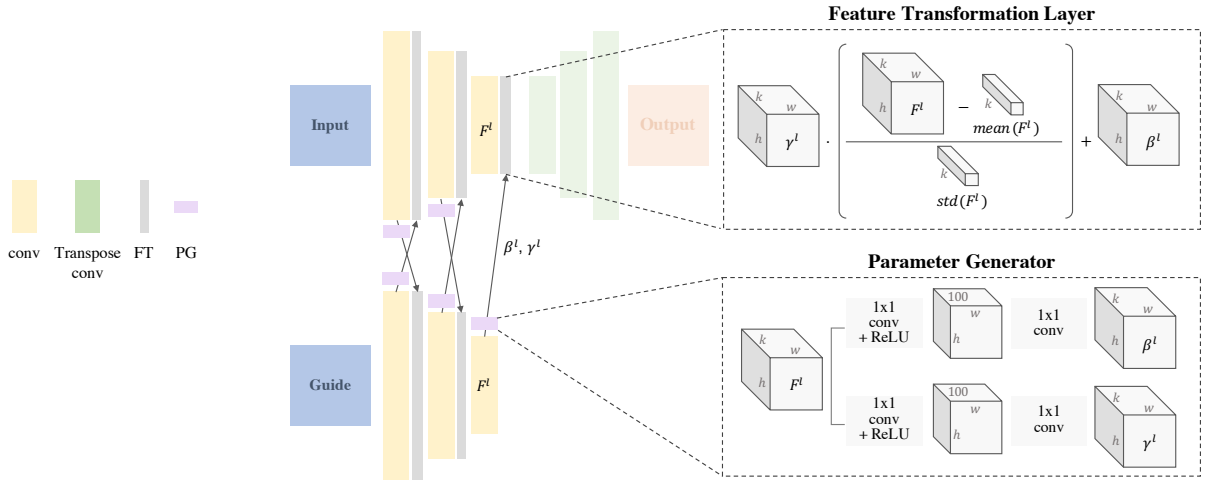


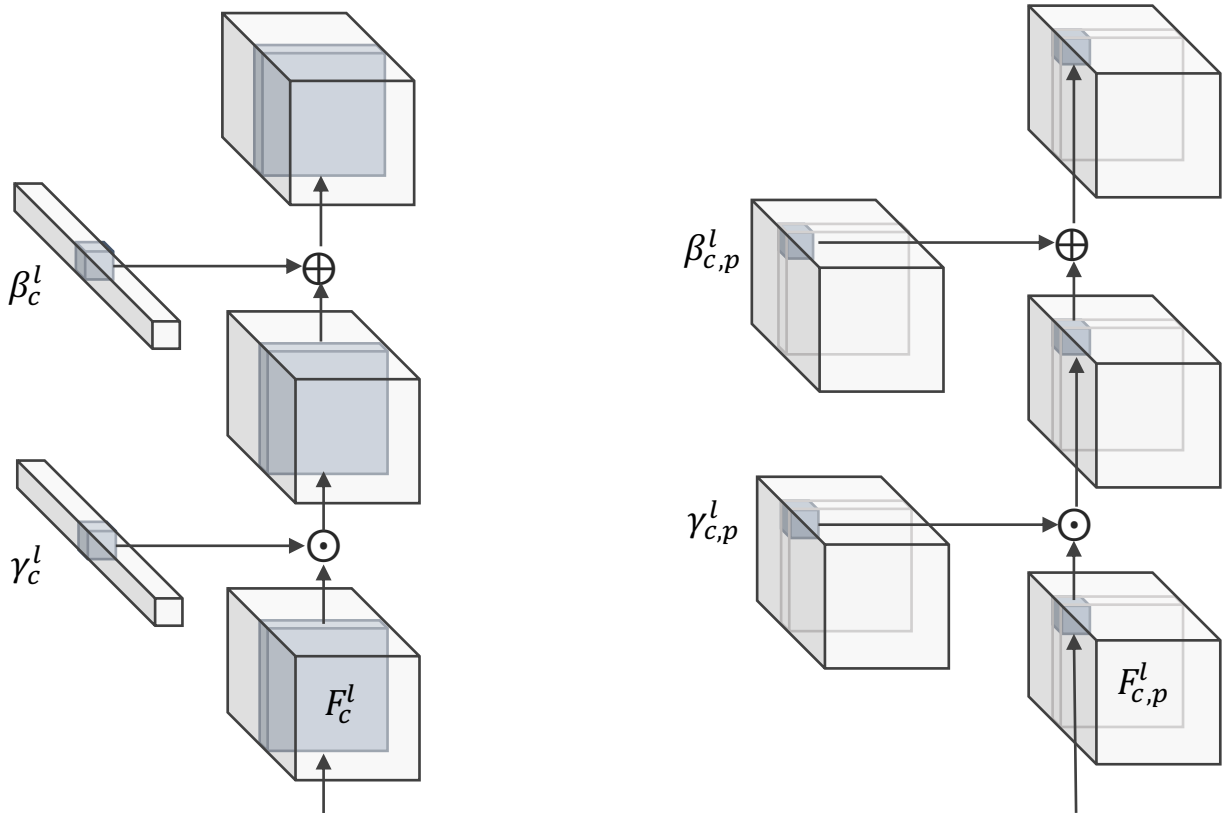
Figure 2.3: **Bi-directional Feature Transformation.** We present a bi-directional feature transformation model to better utilize the additional guidance for guided image-to-image translation problems. In place of every normalization layer in the encoder, we add our novel FT layer. This layer scales and shifts the normalized feature of that layer as shown in Figure 2.4. The scaling and shifting parameters are generated using a parameter generation model of two convolution layers with a bottleneck of 100 dimension.

2.3.1 Feature transformation layer

Here, we first present the feature transformation (FT) layer to incorporate the guidance information. In an FT layer, we perform an affine transformation on the normalized input features using scaling and shifting parameters computed from the features of the given guidance image. In Eqn. 2.1, we show this operation for an l-th layer. The scaling and shifting parameters γ and β are computed from the guidance signal using a *parameter generator* shown in Figure 2.3.

$$F^{l+1}_{\text{input}} = \gamma^l_{\text{guide}} \frac{F^l_{\text{input}} - \text{mean}(F^l_{\text{input}})}{\text{std}(F^l_{\text{input}})} + \beta^l_{\text{guide}}. \tag{2.1}$$

A key difference between the FiLM layer [22] and the proposed FT layer is highlighted in Figure 2.4. Specifically, the scaling γ and shifting β parameters of the FiLM layers are *vectors*



(a) FiLM (b) FT (Ours)

Figure 2.4: **Feature Transformation (FT).** We present a feature transformation layer to incorporate the guidance into the image-to-image translation model. A key difference between a FiLM layer and our FT layer is that the scaling γ and shifting β parameters of the FiLM layer are *vectors*, while in our FT layer they are *tensors*. Therefore, the scaling and shifting operations are applied in spatially varying manner in our FT layer in contrast to spatially invariant modulation as in the FiLM layer.

and are applied channel-wise. That is, the same affine transformation of feature activations is applied the same way regardless of the spatial position on the feature map. Such approaches are reasonable for tasks such as style transfer or visual reasoning. However, they may not be able to capture fine-grained spatial details that are important for image-to-image translation problems. In contrast, the parameters in our FT layer are three-dimensional *tensors* which offer a flexible way for modulating the input features in a spatially varying manner and supports various forms of guidance signals (e.g., dense, sparse, or multi-channel).

2.3.2 Bi-directional conditioning scheme

To further utilize the available information from the guidance image, we propose a *bi-directional conditioning scheme*. Unlike existing conditioning schemes that only allow the guidance signal to influence the input image process, our approach supports bi-directional communication between two branches of the networks processing the input and guidance image. This bi-directional flow of information enables the generative model to better capture the constraints of the guidance image. In our proposed bFT scheme, we replace every normalization layer with our proposed FT layer. At l-th layer, the guidance feature representation manipulates the input feature representation as shown in Eqn. 2.1, and at the same time is manipulated by that input feature representation. Such that:

$$F^{l+1}_{\text{guide}} = \gamma^{l}_{\text{input}} \frac{F^{l}_{\text{guide}} - \text{mean}(F^{l}_{\text{guide}})}{\text{std}(F^{l}_{\text{guide}})} + \beta^{l}_{input} \tag{2.2}$$

Our intuition is that such a bi-directional approach can be seen as a bi-directional communication between a teacher (guidance branch) and a student (input image branch). A one-way communication from the teacher to the student might not help the student understand the teacher as much as two-way communication.

2.4 Experimental Results

We evaluate our proposed bi-directional feature transformation conditioning scheme on three different guided image-to-image translation problems with three different types of the guidance signal.[1] For all tasks, we use GANs with two possible architectures as our generator model, either Unet or Resnet. We follow the same training objective function (a weighted combination of L_1 loss and an adversarial loss L_{GAN}) as in [21]:

$$L_{GAN}(G, D) + \lambda L_1(G). \tag{2.3}$$

where we set λ to 100 for all the experiments. For each task we compare our results with state-of-the-art methods as well as pix2pix [21] (with input concatenation conditioning).

2.4.1 Controllable sketch-to-photo synthesis

In this texture transfer task, given a sketch and a random sized texture patch as the guidance signal, we aim to synthesize a photo that fills the input sketch respecting that given texture patch.

Implementation details We use the Unet architecture of [21] as the base architecture of our model. For both our bFT model and pix2pix, we train using a learning rate of 0.0002 with 7 layers of Unet architecture. We use an Adam optimizer for both with beta1 as 0.5 for pix2pix, and beta1 as 0.9 for our model. For the handbag dataset, we train for 500 epochs with a batch size of 64. For the shoes and clothes datasets, we train for 100 epochs with batch size of 256.

[1] Code available: https://github.com/vt-vl-lab/Guided-pix2pix

Datasets and metrics We use the 128x128 data generated by Xian et al. [1] and follow the same texture patch generation algorithm from the ground truth images. We evaluate the results using the Learned Perceptual Image Patch Similarity (LPIPS) metric proposed by Zhang et al. [17] and the frechet inception distance (FID) proposed by Heusel et al. [18]. For every sketch in the test set, we generate 10 random sized ground truth texture patches using the texture patch generation algorithm from Xian et al. [1] and compute the LPIPS and the FID of the synthesized images. We use the provided pretrained models of Xian et al. [1] to compute their results. Their pretrained models are trained on ground truth patches as well as external patches, while our model and pix2pix are trained only on ground truth patches.

Evaluation We show the quantitative results of our work compared to Isola et al. [21] and Xian et al. [1] in Table 2.1. While our model training is considerably simpler (trained with only two losses) than that of the Xian et al. [1] (with seven different loss terms), we show favorable results against both pix2pix [21] and Xian et al. [1] in terms of the LPIPS metric on all three datasets. We also show the FID results.

Table 2.1: Texture Transfer Task: visual quality evaluation using the Learned Perceptual Image Patch Similarity (LPIPS) metric [17] and Frechet Inception Distance (FID) [18] on the datasets generated by [1]. A lower score is better.

	Handbag Dataset		Shoes Dataset		Clothes Dataset	
	LPIPS	FID	LPIPS	FID	LPIPS	FID
Xian et al. [1]	0.171	60.848	0.124	44.762	0.113	49.568
pix2pix [21]	0.234	96.31	0.238	197.492	0.439	190.161
Ours	0.161	74.885	0.124	121.241	0.067	58.407

We show sample qualitative results on the handbag, shoes, and clothes datasets in Figure 2.5 using ground truth texture patches as the guidance signal.

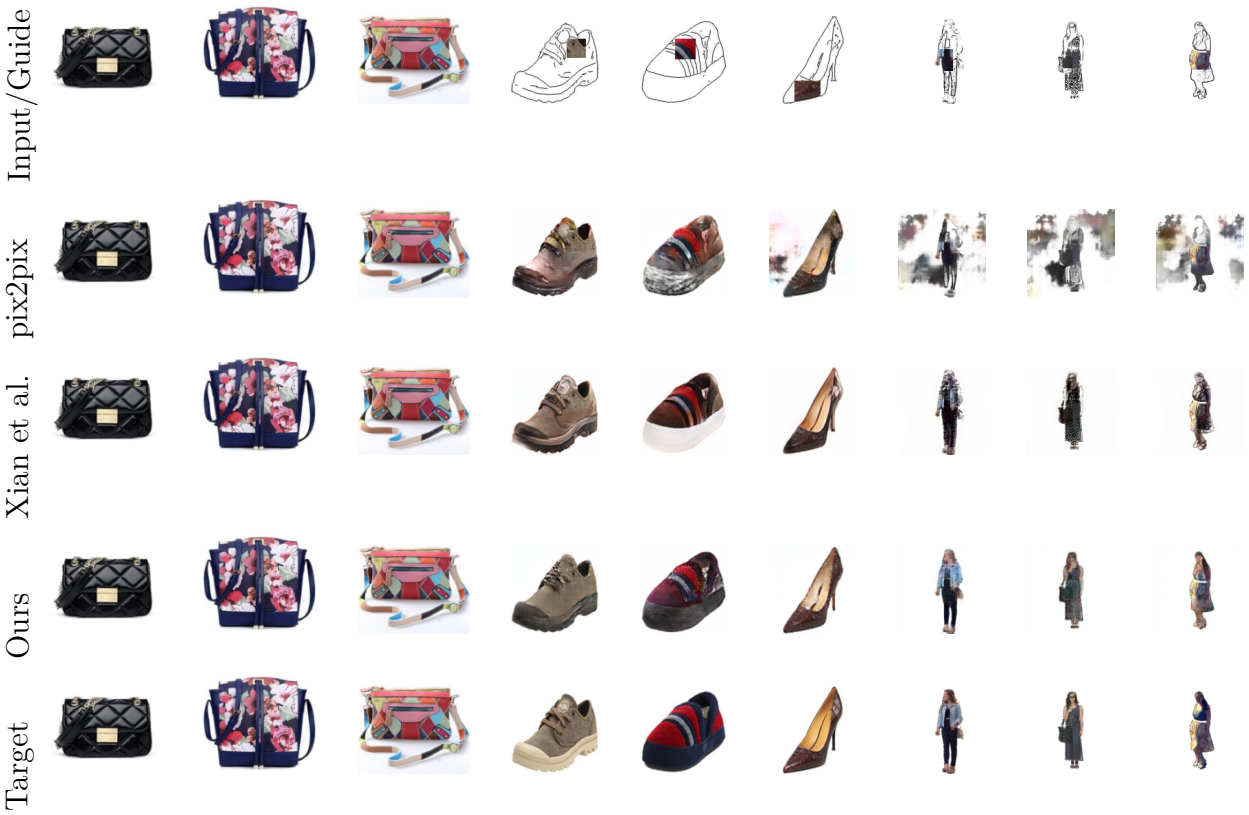


Figure 2.5: **Controllable sketch-to-photo synthesis with texture patches.** Texture transfer qualitative comparison with state-of-the-art-results on the handbags, shoes, and clothes datasets from [1]. Here we use the ground truth texture patches as the guidance signal.

2.4.2 Controllable person-image synthesis

In the pose transfer task, given an image of a person and a target pose as a guidance signal, we aim to synthesize an image of that given person in the desired pose.

Implementation details We use ResNet architecture as the base architecture of our model. For both our bFT model and pix2pix, we train for 100 epochs using a learning rate of 0.0002 with a batch size of 8, then we minimize the learning rate to 0.00002 and train for 50 additional epochs. We use the Adam optimizer for both with beta1 as 0.5 for pix2pix, and beta1 as 0.9 for our model. We use 8 layers for the Unet architecture for pix2pix.

Datasets and metrics We use the 256x256 train and test sets provided by Ma et al. [46] from the DeepFashion dataset [19]. Following the evaluation protocols in literature, we use both SSIM and Inception Score (IS) to measure the quality of the synthesized images. We also use the FID metric.

Evaluation We show the quantitative results of our work compared to state-of-the-art methods in Table 2.2. We note that Siarohin et al. [48] trains on a different training set of the DeepFashion dataset and excludes samples where pose keypoints are not detected. To ensure fair comparison, we modify our test set to exclude such samples. We report the results on both the full test set and the modified one. We use the pretrained models provided by [46, 48] to test their models on our test set. We also note that Siarohin et al. [48] uses the input pose as an additional input to the model. We show favorable results against other methods using the Frechet Inception Distance (FID).

Note that it is very difficult to measure the quality of a synthesized image. In this task, however, we not only care about the quality of the image, but also about it having the same

Table 2.2: Pose Transfer task: visual quality evaluation on the DeepFashion dataset [19]. A higher score of SSIM/IS is better. A lower score of FID is better.

	Full test set			Modified test set		
	SSIM	IS	FID	SSIM	IS	FID
Ma et al. [47]	0.614	3.29	-	-	-	-
Ma et al. [46]	0.762	3.09	47.917	0.764	3.10	47.373
Siarohin et al. [48]	0.758	**3.36**	15.655	0.763	**3.32**	15.215
pix2pix [21]	**0.770**	2.96	66.752	**0.774**	2.93	65.907
Ours	0.767	3.22	**12.266**	0.771	3.19	**12.056**

content and respecting the target pose. We show the qualitative results in Figure 2.6.

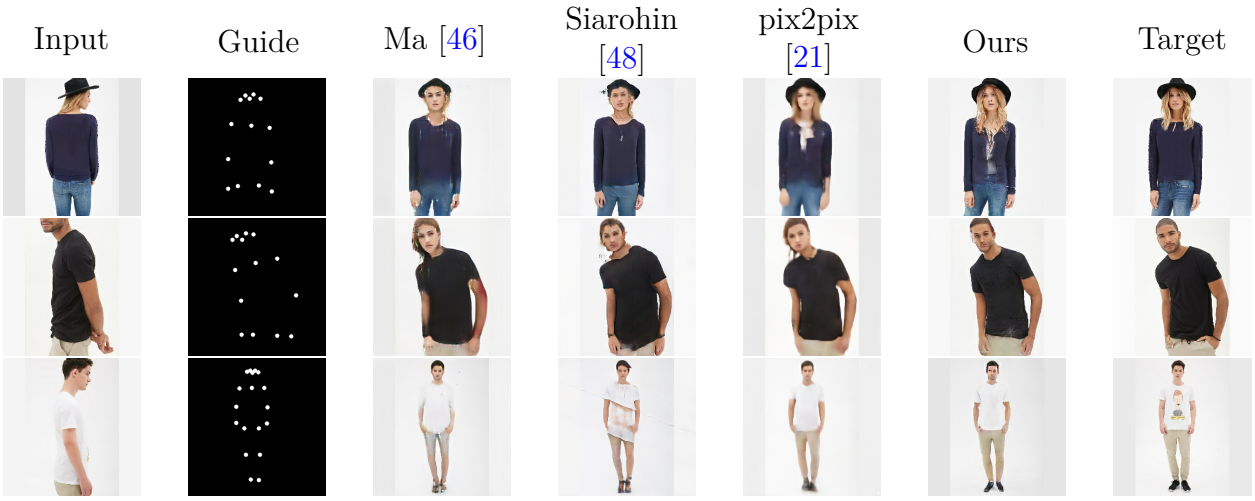


Figure 2.6: **Controllable person-image synthesis with pose keypoints.** Pose transfer qualitative results on DeepFashion dataset. Our model in general achieves sharper results on this challenging task.

Unlike the aforementioned methods that use keypoint based pose, Neverova et al. [49] uses dense pose to perform pose transfer and achieved a score of [SSIM=0.785, IS=3.61], however, we were unable to obtain the data nor the pre-trained model for comparison.

2.4.3 Depth upsampling

In depth upsampling, we aim to generate a high-resolution depth map given a low resolution depth map with the guidance of a high resolution RGB image.

Implementation details We use the ResNet architecture as the base architecture of our model. For both our bFT model and pix2pix, we only use L1 as the objective function and train for 500 epochs using a learning rate of 0.0002 with batch size of 2. We use an Adam optimizer for both with beta1 as 0.5. For our work, we train on the original size of the data 480x640, however, because pix2pix uses square sized inputs, it is trained on 512x512 resized data and we resize back before evaluation. We use 9 layers for the Unet architecture of pix2pix.

Dataset and metric Following the setting of Li et al. [45], we use 1000 samples from the NYU v2 dataset [20] for training and we test on the remaining 449. We generate the low resolution input depth map using bicubic upsampling for three different scale factors 16, 8, and 4. Similar to the works in literature we use RMSE to evaluate the quality of the generated depth.

Evaluation We show the RMSE results of our work compared to Isola et al. [21] and state-of-the-art methods in Table 2.3. We report the results by Li et al. [45].

We also show qualitative results for the three scale factors in Figure 2.7. Our model, while not designed for depth upsampling, can achieve state-of-the-art performance.

Table 2.3: Depth Upsampling task: root mean square error (RMSE) results in centimeters for the NYU v2 dataset [20].

Depth Scale	$x4$	$x8$	$x16$
Bicubic	8.16	14.22	22.32
MRF [52]	7.84	13.98	22.20
GF [53]	7.32	12.98	22.03
JBU [54]	4.07	13.62	22.03
Ham [55]	5.27	12.31	19.24
DMSG [56]	3.48	6.07	10.27
FBS [57]	4.29	8.94	14.59
DJF [58]	3.54	6.20	10.21
DJFR [45]	3.38	5.86	10.11
pix2pix [21]	4.12	6.48	10.17
Ours	**3.35**	**5.73**	**9.01**

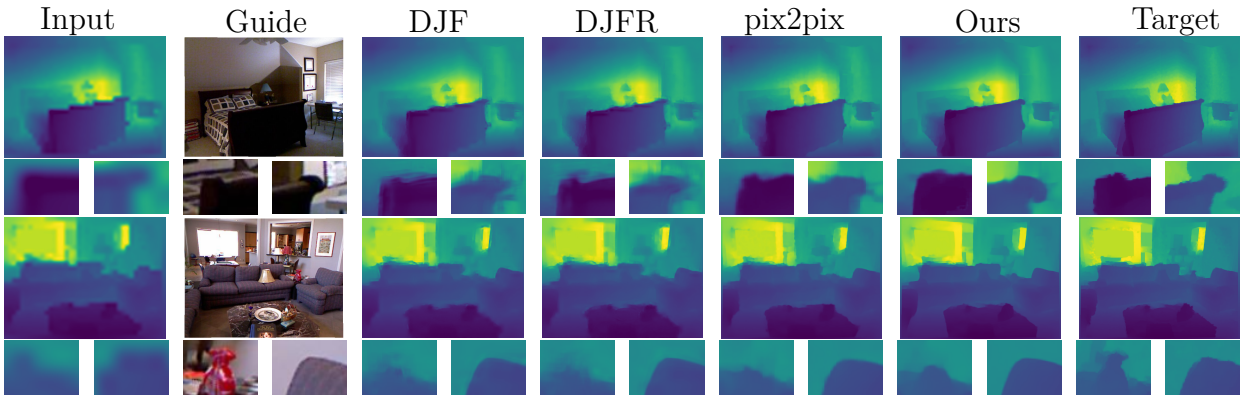


Figure 2.7: **Depth upsampling guided by an RGB image.** Comparison of depth upsampling qualitative results for a scale factor of 16 with the state-of-the-art methods. The zoomed-in crops show that our method is able to capture fine details with sharper edges.

2.4.4 Ablation study

We conduct an ablation study to the effectiveness of our proposed bi-directional conditioning scheme.

Conditioning schemes We compare our proposed bi-directional feature transformation scheme (bFT) to uni-directional feature transformation (uFT), feature concatenation, and input concatenation schemes shown in Figure 2.2. We show quantitative results in Table 2.4.

Table 2.4: Conditioning schemes.

Conditioning method	Depth Upsampling			Pose Transfer			Texture Transfer					
							Handbags		Shoes		Clothes	
	4x	8x	16x	SSIM	IS	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID
Input Concatenation	6.65	8.42	11.86	0.782	3.10	42.330	0.182	85.600	0.137	124.973	0.061	60.795
Feature Concatenation	6.67	7.63	11.59	0.770	3.26	14.672	0.196	87.052	0.145	104.227	0.085	44.900
uFT	5.55	7.26	11.41	0.765	3.18	13.988	0.174	85.273	0.126	119.588	0.071	56.66
bFT (Ours)	3.35	5.73	9.01	0.767	3.17	13.240	0.171	80.179	0.123	119.832	0.067	58.467

Number of feature transformation (FT) layers In our bFT model, we use FT in place of every normalization layer. For pose transfer and depth upsampling tasks, we use a Resnet base with 4 normalization layers. Replacing those layers with our proposed FT layer, we end up with 4 FT layers. We compare our approach with using FT at l, 2, and 3 layers both bi-directionally and uni-directionally. We show the quantitative results in Table 2.5.

Table 2.5: Number of feature transformation (FT) layers.

#Layers	Depth Upsampling		Pose Transfer					
	uFT	bFT	uFT			bFT		
	x16	x16	SSIM	IS	FID	SSIM	IS	FID
1	10.79	10.79	0.786	2.92	59.678	0.786	2.92	59.678
2	10.75	8.96	0.784	2.98	47.411	0.785	3.01	51.458
3	10.26	8.82	0.768	3.15	16.069	0.766	3.24	13.392
4	11.41	9.01	0.765	3.18	13.988	0.767	3.17	13.240

Different approaches to affine transformation Using our bi-directional approach, we compare our proposed FT with CIN and AdaIN. In both CIN and AdaIN, we use FiLM layer in place of every normalization layer. In CIN, we learn the scaling and shifting parameters, while in AdaIN, we use the mean as the scaling parameter and the standard deviation as the shifting parameter. We also test feature transformation at only the last layer of the encoder and compare the performance of our FT with CIN and AdaIN. We show the quantitative results in Table 2.6.

Table 2.6: Different approaches to affine transformation.

Method	Depth Upsampling	Pose Transfer		
	x16	SSIM	IS	FID
Ours	9.01	0.767	3.17	13.240
bi-directional AdaIN	13.36	0.722	3.37	160.846
bi-directional CIN	13.97	0.721	3.36	157.335
Final Layer - FT	11.40	0.769	3.25	18.292
Final Layer - AdaIN	14.30	0.720	3.30	146.596
Final Layer - CIN	14.51	0.720	3.58	168.503

2.4.5 User study

We conduct a user study on pair-wise comparisons. We ask 100 subjects to answer 4 random pair-wise comparisons per task and dataset. We ask the subject to select the image that looks more realistic respecting the input and the given guidance signal. We show the user study results in Figure 2.8.

2.4.6 Limitation

In the task of texture transfer, we observe a limitation of our work when the guidance patch does not go well with the input sketch. In such a case, the color of the guidance patch would

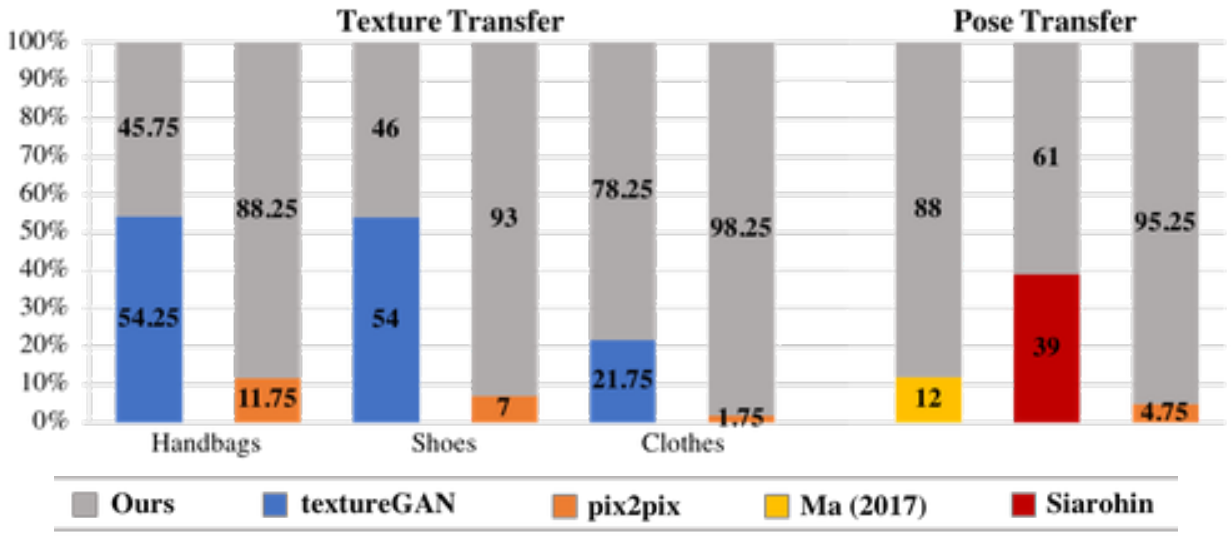


Figure 2.8: **User Study.** The percentage of people that find our method more realistic respecting the input and guidance signal over state-of-the-art methods using pair-wise comparisons.

propagate through the sketch without fully respecting its texture as shown in Figure 2.9.

Figure 2.9: **Failure examples.** When the guided patch does not match well with the given sketch, our model fails to hallucinate the given texture.

2.5 Conclusion

We have presented a new conditional scheme for guided image-to-image translation problems. Our core technical contributions lie in the use of *spatially varying* feature transformation and the design of *bi-directional conditioning* scheme that allow the mutual modulation of the guidance and input network branches. We validate the applicability of our method

on various tasks. While being application-agnostic, our approach achieves competitive performance with the state-of-the-art. The generality of our method opens promising direction of incorporating a wide variety of constraints for image-to-image translation problems.

Chapter 3

Pose with Style: Detail-Preserving Pose-Guided Image Synthesis with Conditional StyleGAN

Figure 3.1: **Detail-preserving pose-guided person image generation.** We present a single-image human reposing algorithm guided by arbitrary body shapes and poses. Our method first transfers local appearance features in the source image to the target pose with a human body symmetry prior. We then leverage a pose-conditioned StyleGAN2 generator with spatial modulation to produce photo-realistic reposing results. Our work enables applications of posed-guided synthesis (*left*) and virtual try-on (*right*). Thanks to spatial modulation, our result preserves the texture details of the source image better than prior work.

We present an algorithm for re-rendering a person from a single image under arbitrary poses. Existing methods often have difficulties in hallucinating occluded contents photo-realistically while preserving the identity and fine details in the source image. We first learn to inpaint the correspondence field between the body surface texture and the source image with a human body symmetry prior. The inpainted correspondence field allows us to transfer/warp local features extracted from the source to the target view even under large pose changes. Directly mapping the warped local features to an RGB image using a simple CNN decoder often leads to visible artifacts. Thus, we extend the StyleGAN generator so that it takes pose as input (for controlling poses) and introduces a spatially varying modulation for the latent space using the warped local features (for controlling appearances). We show that our method compares favorably against the state-of-the-art algorithms in both quantitative evaluation and visual comparison.

3.1 Introduction

Controllable, photo-realistic human image synthesis has a wide range of applications, including virtual avatar creation, reposing, virtual try-on, motion transfer, and view synthesis. Photo-realistic rendering of human images is particularly challenging through traditional computer graphics pipelines because it involves 1) designing or capturing 3D geometry and appearance of human and garments, 2) controlling poses via skeleton-driven deformation of 3D shape, and 3) synthesizing complicated wrinkle patterns for loose clothing. Recent learning-based approaches alleviate these challenges and have shown promising results. These methods typically take inputs 1) a single source image capturing the human appearance and 2) a target pose representation (part confidence maps, skeleton, mesh, or dense UV coordinates) and synthesize a novel human image with the appearance from source and

the pose from the target.

Image-to-image translation based methods [46, 47, 48, 59, 60], building upon conditional generative adversarial networks [21], learn to predict the reposed image from the source image and the target pose. However, as human reposing involves significant spatial transformations of appearances, such approaches often require per-subject training using multiple images from the same persons [37, 61] or are incapable of preserving the person's identity and the fine appearance details of the clothing in the source image.

Surface-based approaches [49, 62, 63, 64] map human pixels in the source image to the canonical 3D surface of the human body (e.g., SMPL model [65]) with part segmentation and UV parameterization. This allows transferring pixel values (or local features) of visible human surfaces in the input image to the corresponding spatial location specified by the target pose. These methods thus retain finer-grained local details and identity compared to image-to-image translation models. However, modeling human appearance as a single UV texture map cannot capture view/pose-dependent appearance variations and loose clothing.

StyleGAN-based methods [6, 9, 66] very recently have shown impressive results for controllable human image synthesis [6, 9] or virtual try-on [66]. The key ingredient is to extend the unconditioned StyleGAN network [2] to a *pose-conditioned* one. While their generated images are photo-realistic, it remains challenging to preserve fine appearance details (e.g., unique patterns/textures of garments) in the source image due to the global (spatially-invariant) modulation/demodulation of latent space.

We present a new algorithm for generating *detail-preserving* and *photo-realistic* re-rendering of human with novel poses from a *single source image*. Similar to the concurrent work [9, 66], we use a pose-conditioned StyleGAN network for generating pose-guided images. To preserve fine-grained details in the source image, we learn to inpaint the correspondence field

between 3D body surface and the source image using a body symmetry prior. Using this inpainted correspondence field, we transfer local features from the source to the target pose and use the warped local features to modulate the StyleGAN generator network at multiple StyleBlocks in a *spatially varying* manner. As we combine complementary techniques of photo-realistic image synthesis (from StyleGAN-based methods) and the 3D-aware detail transfers (from surface-based methods), our method achieves high-quality human-reposing and garment transfer results (Figure 3.1) and alleviates visible artifacts compared with the state-of-the-art (Figure 3.2). While StylePoseGan [9] (concurrent work to ours) also combines pose-conditioned StyleGAN with the use of proxy geometry, its global modulation/demodulation scheme limits its ability to preserve fine appearance details. We evaluate the proposed algorithm visually and quantitatively using the DeepFashion dataset [8] and show favorable results compared to the current best-performing methods. *Our contributions* include:

- We integrate the techniques from surface-based and styleGAN-based methods to produce *detail-preserving* and *photo-realistic* controllable human image synthesis.

- We propose an explicit *symmetry prior* of the human body for learning to inpaint the correspondence field between human body surface and the source image which facilitates detail transfer, particularly for drastic pose changes.

- We present a *spatially varying* variant of latent space modulation in the StyleGAN network, allowing us to transfer local details while preserving photo-realism.

Figure 3.2: **Limitations of existing methods.** Existing human reposing methods struggle to preserve details in the source image. Common issues include identity (1st and 2nd columns) and clothing textures (3rd, 4th columns) changes. Compare these results with ours in Figure 3.1.

3.2 Related Work

3.2.1 Pose-guided Person Image Synthesis

Pose-guided person image synthesis aims to transfer a person's appearance from a source image to a target pose. Example applications include motion transfer [61, 67, 68], human reposing [6, 46, 47], and virtual try-on [6, 9, 66]. These approaches typically encode the pose as either part confidence maps [46, 47, 67] or skeleton [5, 6, 7, 48, 59, 60, 61] and use a conditional GAN to produce the reposed images. To handle large pose changes, existing methods leverage per-subject training [37, 61, 67], spatial transformation/deformation [7, 48, 69], and local attention [7]. To retain the source identity and appearance, surface-based methods first establish the correspondence between pixels from the source/target image to a canonical coordinate system of the 3D human body (with UV parameterization). These methods can then transfer pixels [49, 62, 63, 70] or local features [64] to the target pose. As the commonly used UV parameterization only captures the surface of a tight human body [65], methods either explicitly predict garment labels [68] or implicitly re-render the warped features [64]. Very recently, pose-conditioned StyleGAN networks have been proposed [9, 66]. To control the target appearance, pose-independent UV texture [9] is used to modulate the latent space. Our method builds upon pose-conditioned StyleGAN but differs from prior work in two critical aspects. First, instead of *global* latent feature modulation used in prior work, we propose to use a *spatially varying* modulation for improved local detail transfer. Second, we train a coordinate inpainting network for completing partial correspondence field (between the body surface and source image) using a human body symmetry prior. This allows us to directly transfer local features extracted from the source to the target pose.

3.2.2 Neural Rendering

Neural rendering methods first render a coarse RGB image or neural textures that is then mapped to an RGB image using a translation network [71, 72, 73, 74, 75, 76]. Recent research focuses on learning volumetric neural scene representations for view synthesis [77, 78]. This has been extended to handle dynamic scenes (e.g., humans) [79, 80, 81, 82, 83]. Recent efforts further enable controls over viewpoints [84, 85], pose [86, 87], expressions [85], illumination [88] of human face/body. However, most of these approaches often require computationally expensive *per-scene/per-person training*. Our method also uses body surface mesh as our geometry proxy for re-rendering. Instead of using a simple CNN translation network, we integrate the rendered latent texture with StyleGAN through spatially varying modulation. In contrast to volumetric neural rendering techniques, our method does *not* require per-subject training.

3.2.3 Deep Generative Adversarial Networks

Deep generative adversarial networks have shown great potentials for synthesizing high-quality photo-realistic images [2, 25, 89, 90, 91]. Using the pre-trained model, several works discover directions in the latent space that correspond to spatial or semantic changes [92, 93, 94, 95, 96]. In the context of portrait images, some recent methods provide 3D control for the generated samples [97, 98] or real photographs [99]. Our work focuses on designing a pose-conditioned GAN with precise control on the localized appearance (for virtual try-on) and pose (for reposing).

3.2.4 Image-to-Image Translation

Image-to-image translation provides a general framework for mapping an image from one visual domain to another [21, 26, 100]. Recent advances include learning from unpaired dataset [28, 31, 32], extension to videos [37, 101], and talking heads [102, 103]. Similar to many existing human reposing methods [5, 6, 7, 46, 47, 59, 61], our work maps an input target pose to an RGB image with the appearance from a source image. Our core technical novelties lie in 1) spatial modulation in StyleGAN for detail transfer and 2) a body symmetry prior for correspondence field inpainting.

3.2.5 Localized Manipulation

When editing images, Localized manipulation is often preferable over global changes. Existing work addresses this via structured noise [104], local semantic latent vector discovery [105], latent space regression [106], and explicit masking [107]. Our spatial feature modulation shares high-level similarity with approaches that add spatial dimensions to the latent vectors in unconditional StyleGAN [104, 108] and conditional GANs [100, 109]. Our work differs in that our spatial modulation parameters are predicted from the warped appearance features extracted from the source image instead of being generated from random noise using a mapping network.

3.2.6 Symmetry Prior

Symmetry prior (in particular reflective symmetry) has been applied for learning deformable 3D objects [110], 3D reconstruction of objects [111], and human pose regression [112]. Our work applies left-right reflective symmetry to facilitate the training of coordinate-based tex-

ture inpainting network. The symmetry prior allows us to reuse local appearance features from the source and leads to improved results when source and target poses are drastically different.

3.3 Method

Given an image of a person I_{src} and a desired target pose P_{trg} represented by Image-space UV coordinate map per body part (shortly IUV) extracted from DensePose [3], our goal is to generate an image preserving the appearance of the person in I_{src} in the desired pose P_{trg}. Note that this IUV representation of dense pose entangles both the pose and shape representation.

We show an overview of our proposed approach in Figure 3.3.

We use a pose-guided StyleGAN2 generator [2] that takes $16 \times 16 \times 512$ pose features F_{pose} as input. The pose features F_{pose} are encoded from the DensePose representation [3] using a pose feature generator G_{pose} that is composed of several residual blocks. Using the source and target pose, we use coordinate completion model to produce target coordinate that establishes the correspondence between target and source image. To encode the appearance information, we use a feature pyramid network [4] G_{app} to encode the source image into multiscale features and warp them according to the target pose. We then use the warped appearance features to generate scaling and shifting parameters to spatially modulate the latent space of the StyleGAN generator.

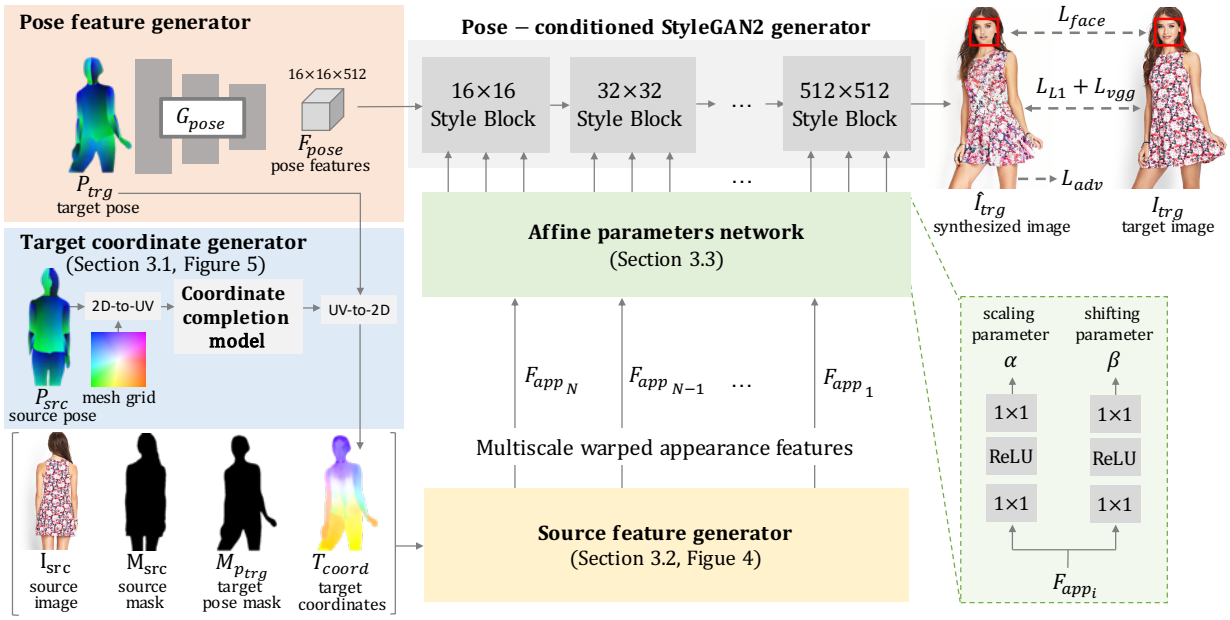


Figure 3.3: **Method overview.** Our human reposing model builds upon a pose-conditioned StyleGAN2 generator [2]. We extract the DensePose [3] representation P_{trg} and use a pose encoder G_{pose} to encode P_{trg} into $16 \times 16 \times 512$ pose features F_{pose} which is used as input to the StyleGAN2 generator [2]. To preserve the source image appearance, we encode the input source image I_{src} into multiscale warped appearance features F_{app_i} using the source feature generator (Figure 3.7). To warp the feature from the source pose to the target pose we use the target coordinates T_{coord}. We compute these target coordinates T_{coord} using 1) the target dense pose P_{trg} and 2) the completed coordinates in the UV-space inpainted using the coordinate completion model (Figure 3.4). We pass the multi-scale warped appearance features F_{app_i} through the affine parameters network to generate scaling and shifting parameters α and β that are used to modulate the StyleGAN2 generator features in a *spatially varying* manner (Figure 3.6). Our training losses include adversarial loss, reconstruction losses, and a face identity loss.

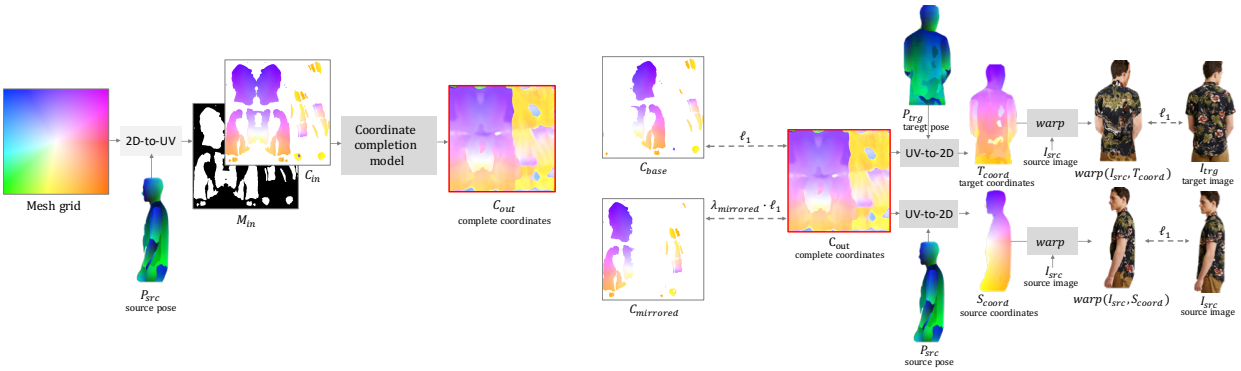
3.3.1 Coordinate Completion Model

The IUV map of the source pose P_{src} allows us to represent the *pose-independent* appearance of the person in the UV-space. However, only the appearance of *visible* body surface can be extracted. This leads to incomplete UV-space appearance representation and thus may not handle the dis-occluded appearance for the target pose P_{trg}. Previous work [9] encodes the partial UV-space appearance to a *global latent vector* for modulating the generator. This works well for clothing with uniform colors or homogeneous textures, but inevitably loses the spatially-distributed appearance details. We propose to inpaint the UV-space appearance by a neural network guided by the human body mirror-symmetry prior. Instead of directly inpainting pixel values in UV-space, we choose to complete the mapping from image-space to UV-space established by P_{src} and represented by UV-space source image coordinates, in order to avoid generating unwanted appearance artifacts while best preserving the source appearance. We refer to this network as *coordinate completion model*. We show an overview of our coordinate completion model in Figure 3.4.

Given a mesh grid and the dense pose of the input source image P_{src}, we use a pre-computed image-space to/from UV-space mapping to map coordinates from the mesh grid to appropriate locations in the UV-space (using bilinear sampling for handling fractional coordinates). We denote these base mapped coordinates as C_{base} and the mask indicating where these coordinates are as M_{base}.

Since human appearances are often left-right symmetrical, in addition to these base coordinates, we also map the left-right mirrored coordinates to the UV-space $C_{mirrored}$ and denote their respective mask as $M_{mirrored}$. Visualization of mapped base and mirrored coordinates are shown in Figure 3.5.

We combine the incomplete UV-space base and mirrored coordinates and their respective



(a) Coordinate completion model (b) Training losses

Figure 3.4: **Coordinate completion model.** The goal of the coordinate completion model is to learn how to *reuse* the local features of the visible parts of the human in the source image for the invisible parts (unseen in the source pose) in the target pose. (a) Given a mesh grid and the dense pose of the input source image P_{src}, we map the base coordinates C_{base} and their symmetric counterpart $C_{mirrored}$ from the 2D mesh grid to the UV-space using a pre-computed mapping table. We then concatenate the combined coordinates C_{in} and their corresponding visibility mask M_{in} as input to the coordinate completion model. (b) We train the model to minimize the L1 loss between the predicted coordinates C_{out} and the input coordinates C_{in} as shown in Eqn. 3.4. We also minimize the L1 loss between the warped source image and the warped target image as shown in Eqn. 3.5.

masks, such that:

$$M_{mirrored} = M_{mirrored} - (M_{base} \cdot M_{mirrored}) \tag{3.1}$$

$$M_{in} = M_{base} + M_{mirrored} \tag{3.2}$$

$$C_{in} = C_{base} \cdot M_{base} + C_{mirrored} \cdot M_{mirrored} \tag{3.3}$$

We concatenate the combined coordinates C_{in} and their mask M_{in} and pass them as input to the coordinate completion model. To implement our coordinate completion model, we follow a similar architecture to the coordinate inpainting architecture proposed by [62] with gated convolutions [50].

We train our model to minimize the ℓ_1 loss between the generated coordinates C_{out} and the

input coordinates, such that:

$$\begin{aligned} L_{coord} &= ||C_{out} \cdot M_{base} - C_{base} \cdot M_{base}||_1 \\ &+\lambda_{mirrored} \cdot ||C_{out} \cdot M_{mirrored} - C_{mirrored} \cdot M_{mirrored}||_1, \end{aligned} \tag{3.4}$$

where $\lambda_{mirrored}$ is set to 0.5.

We also utilize the source-target pairs to train the coordinate completion model. Specifically, we use the source dense pose P_{src} to map the generated complete coordinates from the UV-space to the source image-space S_{coord}. Similarly, we also use the target dense pose P_{trg} to map the generated complete coordinates from the UV-space to the target image-space T_{coord} using the pre-computed mapping table. We then use these target and source coordinates to warp pixels from the input source image I_{src} and minimize the ℓ_1 loss between the foreground of the warped images and the foreground of the ground truth images, such that:

$$\begin{aligned} L_{rgb} &= ||warp(I_{src}, S_{coord}) \cdot M_{P_{src}} - I_{src} \cdot M_{P_{src}}||_1 \\ &+||warp(I_{src}, T_{coord}) \cdot M_{P_{trg}} - I_{trg} \cdot M_{P_{trg}}||_1, \end{aligned} \tag{3.5}$$

where $M_{P_{src}}$ and $M_{P_{trg}}$ are the source pose mask and target pose mask, respectively.

The total loss to train the coordinate completion model is:

$$L = L_{coord} + L_{rgb} \tag{3.6}$$

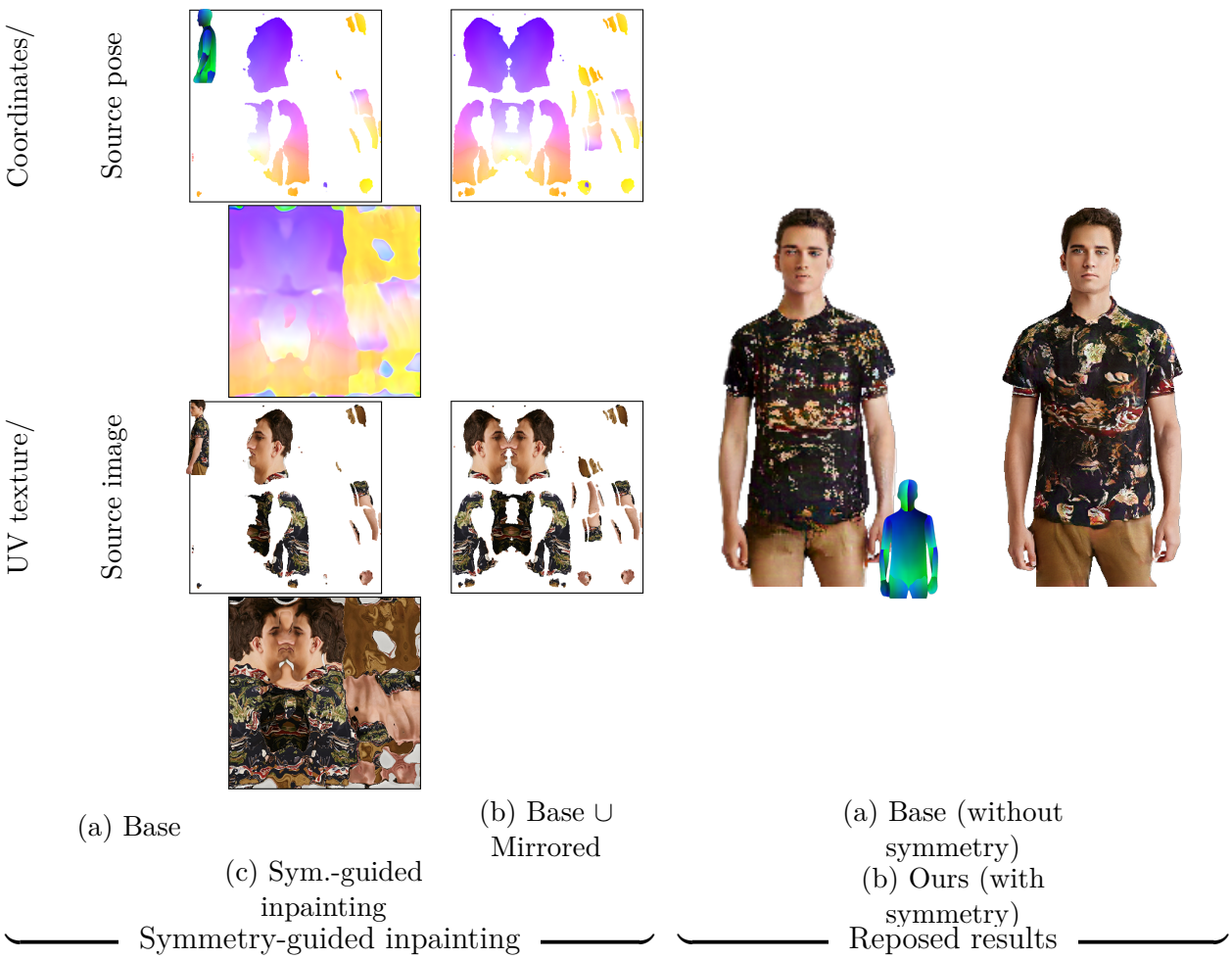
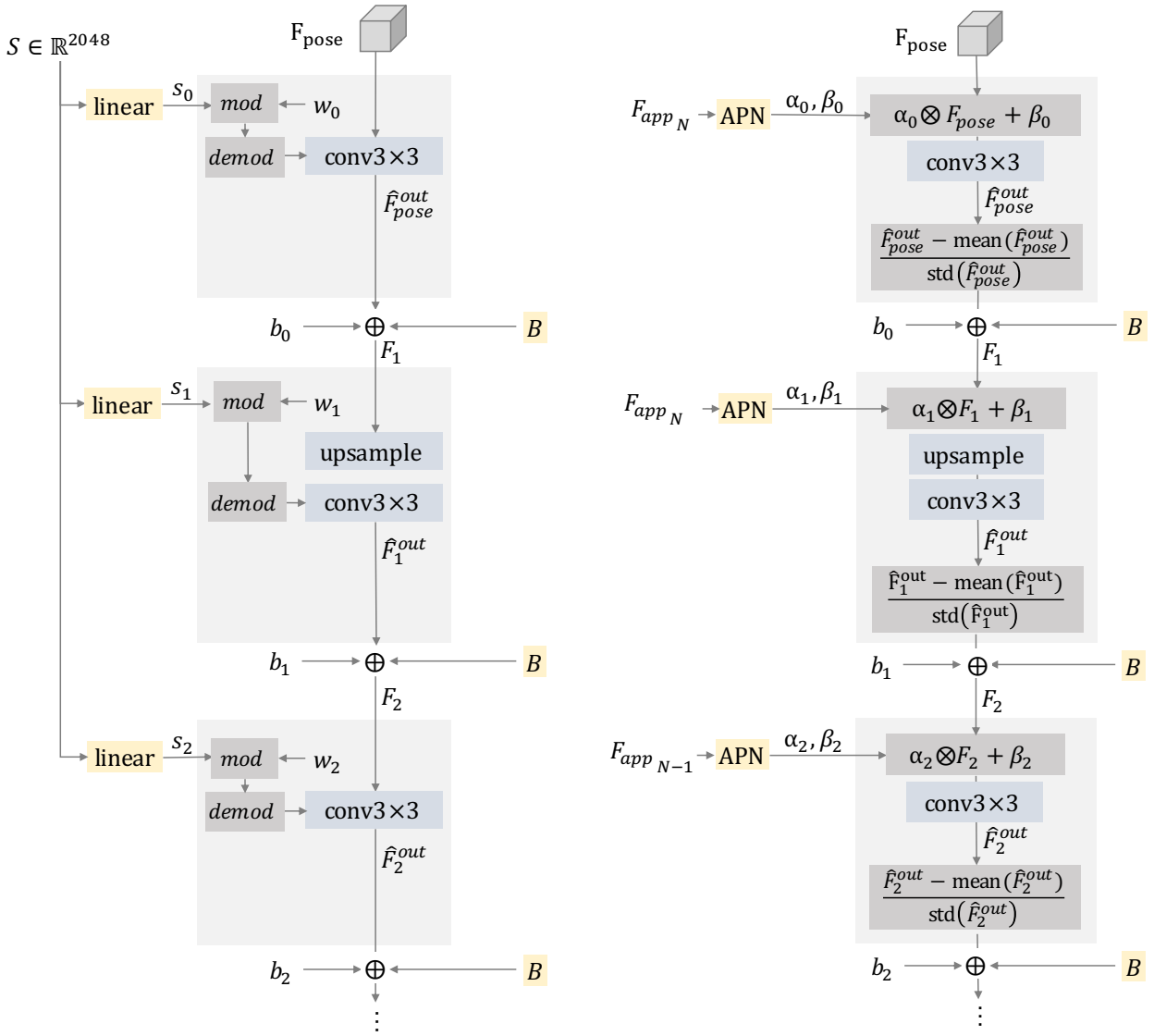


Figure 3.5: **Symmetry-guided inpainting.** (*Left*) Given a mesh grid and the source image dense pose P_{src}, we first map the coordinates from the 2D mesh grid to appropriate locations in the UV-space using a pre-computed mapping table. We can then use these mapped base coordinates C_{base} to warp RGB pixels from the input source image I_{src}. We show the base coordinates and their warped RGB pixels in (a). (b) In addition to these base coordinates, we can also map the left-right *mirrored* coordinates $C_{mirrored}$ from the 2D mesh grid to the UV space. To train our coordinate completion model, we combine the incomplete base and mirrored coordinates in the UV-space. We then concatenate these combined coordinates with their respective mask and pass them as input to our coordinate completion model. We show our completed coordinates and the UV texture map in (c). (*Right*) We compare the reposing results *without* and *with* the proposed symmetry prior.



(a) Non-spatial modulation (StyleGAN2)

(b) Spatial modulation (Ours)

Figure 3.6: **Spatial vs. non-spatial modulation of StyleGAN2 features.** Our input to the StyleGAN2 generator is the encoded target pose features F_{pose} (a) StyleGAN2 [2] performs *non-spatial modulation* of features by modulating and demodulating the weights of the convolutions using the learned style vector S. After the convolution the bias is added as well as StyleGAN2 noise broadcast operation B. (b) To better leverage the spatial features for preserving appearance details, we propose *spatial modulation* of styleGAN2 features. Instead of modulating and demodulating the weights of the convolutions, we modulate the mean and standard deviation of the features. We perform this modulation before the convolution using the shifting and scaling parameters, α and β, generated by the affine parameters network (APN). We then normalize the output of the convolution to zero mean and unit standard deviation before adding the bias and StyleGAN2 noise broadcast operation B.

3.3.2 Source Feature Generator

To preserve the appearance in source image I_{src}, we encode it using several residual blocks into multi-scale features $F^{src}_{app_i}$. We utilize the pretrained coordinate completion model to obtain the target image-space coordinates T_{coord} such that it could warp the source features $F^{src}_{app_i}$ from the source pose to the target pose $F^{trg}_{app_i}$. We then concatenate these warped features with the target dense pose mask $M_{p_{trg}}$ and pass them into a feature pyramid network [4] to get our multi-scale warped appearance features F_{app_i}. We show our source feature generator in Figure 3.7.

3.3.3 Affine Parameters Network and Spatial Modulation

Prior to every convolution layer in each style block of StyleGAN2, we pass the warped source features F_{app_i} into an affine parameters network to generate scaling α, and shifting β parameters. Each convolution layer has its own independent affine parameters network which is composed of two 1×1 convolutions separated by a ReLU activation function for each parameter. To preserve spatial details, we modify every convolution layer in each style block of StyleGAN2 [2]. Instead of performing spatially invariant weight modulation and demodulation, we use the generated scaling and shifting tensor parameters, α and β, to perform spatially varying modulation of the features F_i as follows:

$$\hat{F}_{\mathrm{i}} = \alpha_{\mathrm{i}} \otimes F_{\mathrm{i}} + \beta_{\mathrm{i}}, \tag{3.7}$$

where $\hat{F}_{\mathrm{i}}$ is now the modulated features that will be passed as input to the 3×3 convolution of styleGAN2 generator. The output features of the convolution $\hat{F}_{\mathrm{i}}^{out}$ is then normalized to

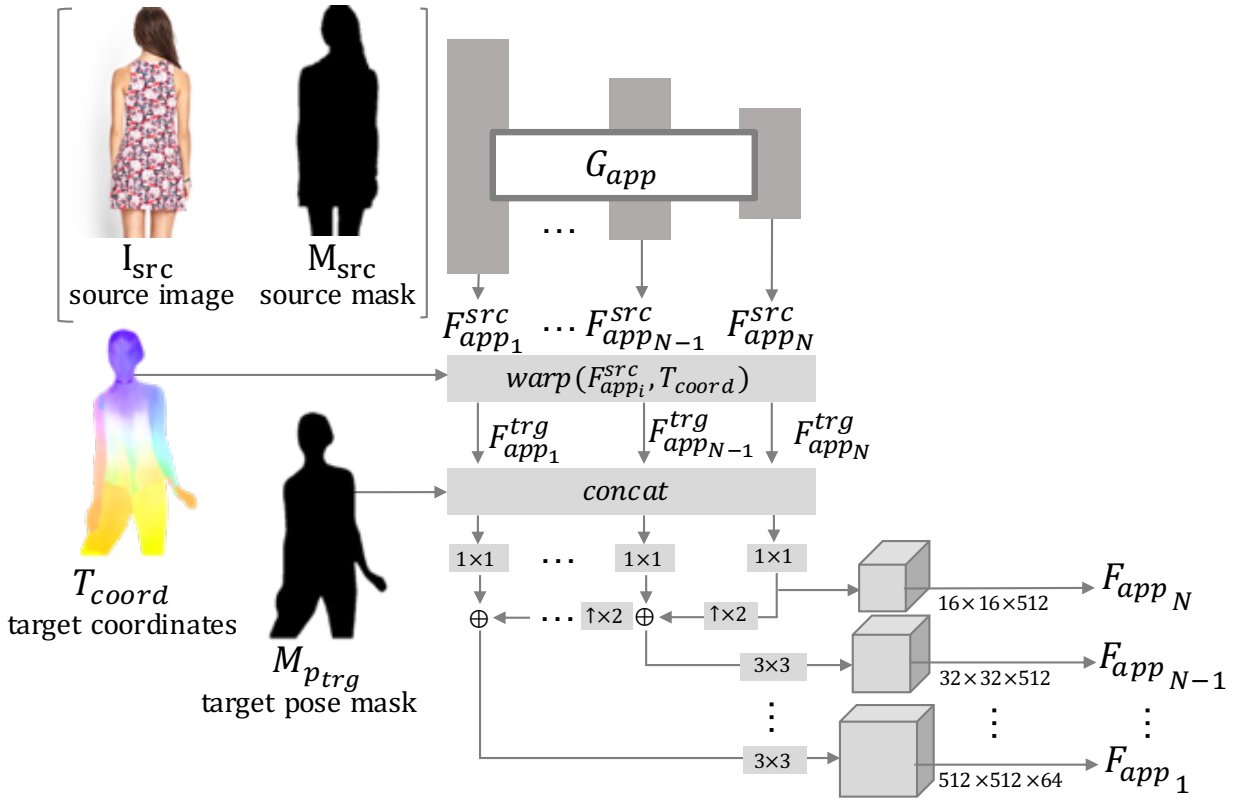


Figure 3.7: **Source feature generator.** To preserve the source image appearance, we encode the input source image I_{src} into multiscale features $F^{src}_{app_i}$ and warp them from the source pose to the target pose $F^{trg}_{app_i}$ using the target coordinates T_{coord} computed from the target coordinate generator (Figure 3.3). We further process the warped features with a feature pyramid network [4] to obtain the multi-scale warped appearance features F_{app_i} which go through affine parameters network to generate scaling and shifting parameters α and β that are used to modulate the StyleGAN2 generator features in a *spatially varying* manner (Figure 3.6).

zero mean and unit standard deviation. Such that:

$$\bar{F}_{\mathrm{i}} = \frac{\hat{F}_{\mathrm{i}}^{out} - \mathrm{mean}(\hat{F}_{\mathrm{i}}^{out})}{\mathrm{std}(\hat{F}_{\mathrm{i}}^{out})} \tag{3.8}$$

Similar to StyleGAN2, we then add the noise broadcast operation B and the bias to $\bar{F}_{\mathrm{i}}$ to get F_{i+1} which will be fed into the next convolution layer. In Figure 3.6, we illustrate our detailed spatial modulation approach as well as the non-spatial weight modulation and demodulation of StyleGAN2.

3.3.4 Training Losses

In addition to StyleGAN2 adversarial loss L_{adv}, we train our model to minimize the following reconstruction losses:

- ℓ_1 loss. We minimize the ℓ_1 loss between the foreground human regions of the synthesized image $\hat{I}_{trg}$ and of the ground truth target I_{trg}.

$$L_{\ell_1} = ||\hat{I}_{trg} \cdot M_{trg} - I_{trg} \cdot M_{trg}||_1, \tag{3.9}$$

  where M_{trg} is the human foreground mask estimated using a human parsing method [113].

- Perceptual loss. We minimize the weighted sum of the ℓ_1 loss between the pretrained VGG features of the synthesized $\hat{I_{trg}}$ foreground and the ground truth I_{trg} foreground such that:

$$L_{vgg} = \sum_{i=1}^{5} w_i \cdot ||VGG_{l_i}(\hat{I}_{trg} \cdot M_{trg}) - VGG_{l_i}(I_{trg} \cdot M_{trg})||_1 \tag{3.10}$$

We use $w = [\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, 1.0]$ and VGG ReLU output layers $l = [1, 6, 11, 20, 29]$ following [26, 100].

- Face identity loss. We use MTCNN [114] to detect, crop, and align faces from the generated image $\hat{I}_{trg}$ and ground truth target I_{trg}. When a face is detected, we maximize the cosine similarity between the pretrained SphereFace [115] features of the generated face and the ground truth target face. Such that:

$$L_{face} = 1 - \left(\frac{SF(\hat{I}_{trg})^{\top} SF(I_{trg})}{\max(||SF(\hat{I}_{trg})||_2 \cdot ||SF(I_{trg})||_2, \epsilon)} \right) \tag{3.11}$$

  where SF is the pretrained SphereFace feature extractor, and $SF(\hat{I}_{trg})$ and $SF(I_{trg})$ are features of the aligned faces of the generated and ground truth image respectively. $\epsilon = e^{-8}$ is a very small value to avoid zero-dividing.

Therefore, our final loss is: $L = L_{adv} + L_{\ell_1} + L_{vgg} + L_{face}$.

3.4 Experimental Results

3.4.1 Experimental setup

Implementation details

We implement our model with PyTorch. We use ADAM optimizer with a learning rate of $ratio \cdot 0.002$ and beta parameters $(0, 0.99^{ratio})$. We set the generator ratio to $\frac{4}{5}$ and discriminator ratio to $\frac{16}{17}$.

Training

We first train our model by focusing on generating the foreground. We apply the reconstruction loss and the adversarial loss only on the foreground. We set the batch size to 1 and train for 50 epochs. This training process takes around 7 days on 8 NVIDIA 2080 Ti GPUs. We then finetune the model by applying the adversarial loss globally on the entire image. We set the batch size to 8 and train for 10 epochs. This training process takes less than 2 days on 2 A100 GPUs. At test time, generating a reposing results with 384×512 resolution takes 0.4 seconds using 1 NVIDIA 2080 Ti GPU.

Dataset

We use the DeepFashion dataset [8] for training and evaluation. We follow the train/test splits (101,967 training and 8,570 testing pairs) of recent methods [5, 6, 7].

3.4.2 Evaluations

Quantitative evaluation

is reported in Table 3.1. We report the human foreground peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), learned perceptual image patch similarity (LPIPS) [116], and Frechet Inception Distance (FID) [117]. PSNR/SSIM often do not correlate well with perceived quality, particularly for synthesis tasks. For example, PSNR may favor blurry results over sharp ones. We report these metrics only for completeness.

Our method compares favorably against existing works such as PATN [5] ADGAN [6], and GFLA [7]. Our method also compares favorably against the concurrent work Style-PoseGAN [9]. We report the quantitative evaluation in Table 3.2. We train and test our

Table 3.1: Quantitative comparison with the state-of-the-art methods on the DeepFashion dataset [8].

	PSNR↑	SSIM↑	FID↓	LPIPS↓
Resolution = 174 × 256				
PATN [5]	17.7041	0.7543	21.8568	0.195
ADGAN [6]	17.7223	0.7544	16.2686	0.175
GFLA [7]	18.0424	0.7625	15.1722	0.167
Ours	**18.5062**	**0.7784**	**8.7449**	**0.134**
Resolution = 348 × 512				
GFLA [7]	17.9718	0.7540	18.8519	0.170
Ours	**18.3567**	**0.7640**	**9.0002**	**0.143**

method using their DeepFashion dataset train/test split.

Table 3.2: Quantitative comparison on 348 × 512 resolution with StylePoseGAN [9] on their DeepFashion dataset train/test split.

	PSNR↑	SSIM↑	FID↓	LPIPS↓
StylePoseGAN [9]	17.7568	0.7508	7.4804	0.167
Ours	**18.5029**	**0.7711**	**6.0557**	**0.144**

Visual comparison

in Figure 3.8 and Figure 3.9 show that our proposed approach captures finer-grained appearance details from the input source images.

Face identity

We evaluate our model's ability to preserve the reposed person's identity. For test images with visible faces (7,164 from 8,570), we report in Table 3.3 the averaged cosine similarity between the face features (Arcface [118] and Spherenet [119]) extracted from the aligned

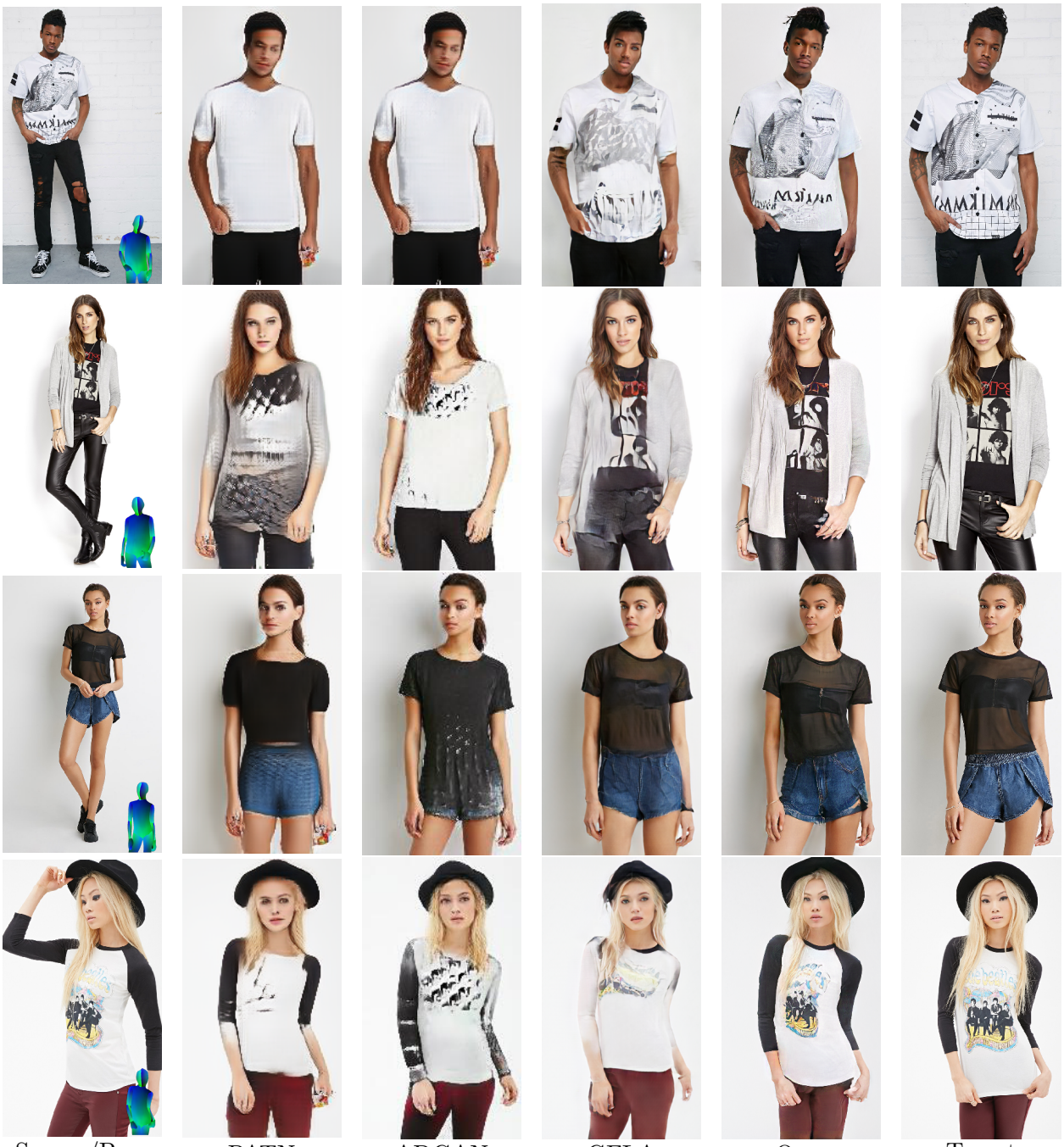


Figure 3.8: **Visual comparison for human reposing.** We show visual comparison of our method with PATN [5], ADGAN [6], and GFLA [7] on DeepFashion dataset [8]. Our approach successfully captures the local details from the source image.

Figure 3.9: **Visual comparison for human reposing.** We compare our method with StylePoseGAN [9] on their train/test split of DeepFashion dataset [8]. Our approach preserves the appearance and captures the fine-grained details of the source image.

faces in the source/target images. Our method compares favorably against GFLA [7].

Table 3.3: The ability to preserve the identity of the reposed person.

	Arcface	Spherenet
GFLA [7]	0.117	0.197
Ours	0.373	0.438
Ground truth	**0.555**	**0.444**

3.4.3 Ablation study

For the ablation study, we report the results of the foreground-focused trained model.

Symmetry prior

We evaluate the effectiveness of adding the symmetry prior to the input of the coordinate completion model. We train our networks *with* and *without* this symmetry prior and report the quantitative results in Table 3.4. Results show that adding the symmetry prior indeed improves the quality of the synthesis. We note that the symmetry prior generally works well for repetitive/textured patterns, but may introduce artifacts for unique patterns (e.g., text).

Table 3.4: The effect of symmetry-guided coordindate inpainting on the DeepFashion dataset [8].

	PSNR↑	SSIM↑	FID↓	LPIPS↓
Without symmetry	18.8810	0.7886	8.5240	0.129
With symmetry (Ours)	**18.9657**	**0.7919**	**8.1434**	**0.124**

Modulation schemes

We quantitatively and qualitatively demonstrate the effectiveness of our proposed spatial modulation. We show quantitative evaluation in Table 3.5. We show qualitative results in Figure 3.10. Results show that spatially varying modulation improves the quality of the synthesized human foreground and captures the spatial details of the source images regardless of the input source image type.

Table 3.5: Ablation on source for appearance (Incomplete UV, Complete UV, and Image) and modulation types (Spatial and Non-spatial).

ID	Input Source	Spatial	PSNR↑	SSIM↑	FID↓	LPIPS↓
A	Incomplete UV	-	18.7385	0.7710	9.4150	0.151
B	Incomplete UV	✓	18.6005	0.7696	9.2321	0.146
C	Complete UV	-	**18.9407**	**0.7770**	9.7435	0.147
D	Complete UV	✓	18.7063	0.7720	<u>9.0236</u>	<u>0.143</u>
E	Source image	-	18.6027	0.7678	9.4367	0.154
F	Source image	✓	<u>18.7420</u>	<u>0.7739</u>	**8.8060**	**0.139**

Sources of appearance

We experiment with multiple variants for encoding source appearance. Specifically, the *incomplete UV*, *complete UV* (completed using the coordinate completion model), and *source image* (our approach shown in Figure 3.3). We report the quantitative results in Table 3.5 and visual results in Figure 3.10. The results show that extracting appearance features directly from the source image preserves more details than other variants.

3.4.4 Garment transfer Results

Using the UV-space pre-computed mapping table, we can segment the UV-space into human body parts (Figure 3.11). We can then use this UV-space segmentation map to generate the

Figure 3.10: **Ablation.** We compare our results with other variants, including the modulation types and source of appearance features. We show that the proposed spatial modulation captures finer-grained details from the source image. Transferring appearance features from the source image leads to fewer artifacts compared to features from the UV space.

target pose segmentation map using the target dense pose P_{trg}. The target segmentation map allows us to combine partial features from multiple source images to perform garment transfer. Figure 3.12 shows examples of bottom and top garment transfer.

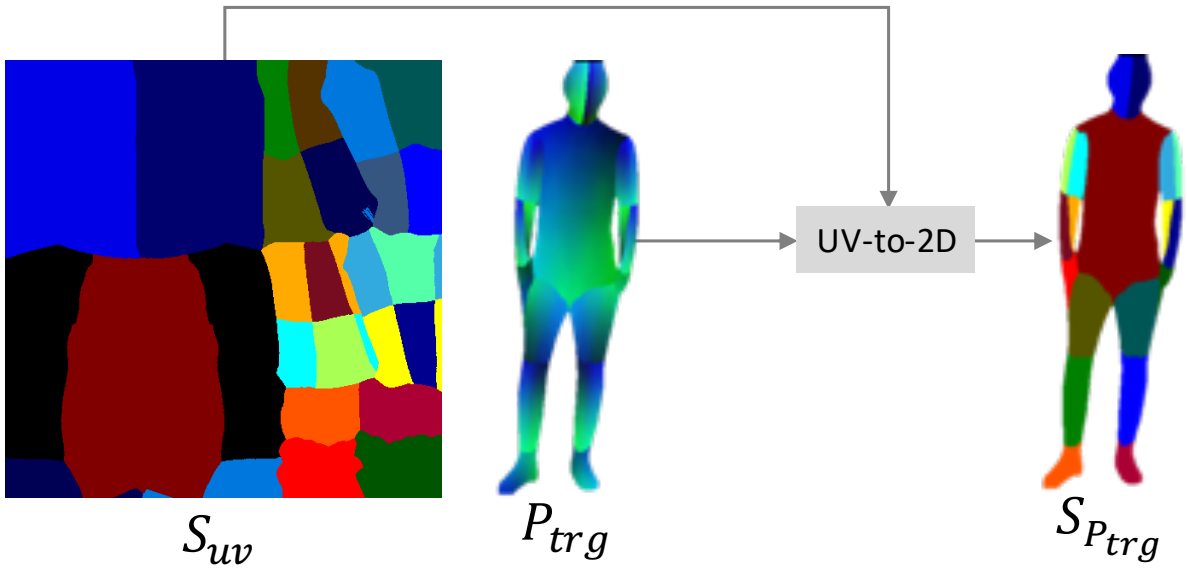


Figure 3.11: **Human body segmentation.** We create a UV-space segmentation map S_{uv} of the human body using the UV-space pre-computed mapping table. We use the target dense pose P_t to map this UV-space segmentation map to 2D target pose S_{P_t} which can then be used to combine features from multiple source images to perform garment transfer.

3.4.5 Limitations

Failure cases

Human reposing from a single image remains challenging. Figure 3.13 shows two failure cases where our approach fails to synthesize realistic hands and clothing textures. Hands are difficult to capture due to the coarse granularity of DensePose. Explicitly parsing hands could help establish more accurate correspondence between source/target pose (e.g., using Monocular total capture [120]). Long-hairs and loose-fit clothes (e.g., skirts) are challenging because they are not captured by DensePose. We believe that incorporating human/garment parsing in our framework may help mitigate the artifacts.

Figure 3.12: **Garment transfer.** We show examples of garment transfer for bottom (*left*) and top (*right*) garment sources.

Figure 3.13: **Failure cases.** Our method produces artifacts on the hands (*left*) and the skirt (*right*).

Diversity

DeepFashion dataset consists of mostly young fit models and very few dark-skinned individuals. Our trained model thus inherit the biases and perform worse on unrepresented individuals as shown in Figure 3.14. We believe that training and evaluating on diverse populations and appearance variations are important future directions.

3.5 Conclusions

We have presented a simple yet effective approach for pose-guided image synthesis. Our core technical novelties lie in 1) spatial modulation of a pose-conditioned StyleGAN generator and 2) a symmetry-guided inpainting network for completing correspondence field. We demonstrate that our approach is capable of synthesizing *photo-realistic* images in the desired target pose and *preserving details* from the source image. We validate various design choices

Figure 3.14: **Diverse in the wild cases.** Our model inherits the biases of DeepFashion dataset and thus performs worse on unrepresented individuals. Our method cannot accurately synthesize curly hair (*left*) and fails in reposing dark-skinned individuals (*right*).

through an ablation study and show improved results when compared with the state-of-the-art human reposing algorithms. Our controllable human image synthesis approach enables high-quality human pose transfer and garment transfer, providing a promising direction for rendering human images.

Chapter 4

Temporally consistent semantic video editing

Figure 4.1: **Temporally consistent video semantic editing.** We present a method for editing the semantic attributes of a video using a pre-trained StyleGAN model. Here we showcase free-form text based editing from SytleCLIP [10] to make the person appear “angry” (2nd row) or wear “eyeglasses” (3rd row).

Generative adversarial networks (GANs) have demonstrated impressive image generation quality and semantic editing capability of real images, e.g., changing object classes, modifying

attributes, or transferring styles. However, applying these GAN-based editing to a video independently for each frame inevitably results in temporal flickering artifacts. We present a simple yet effective method to facilitate temporally coherent video editing. Our core idea is to minimize the temporal photometric inconsistency by optimizing both the latent code and the pre-trained generator. We evaluate the quality of our editing on different domains and GAN inversion techniques and show favorable results against the baselines.

4.1 Introduction

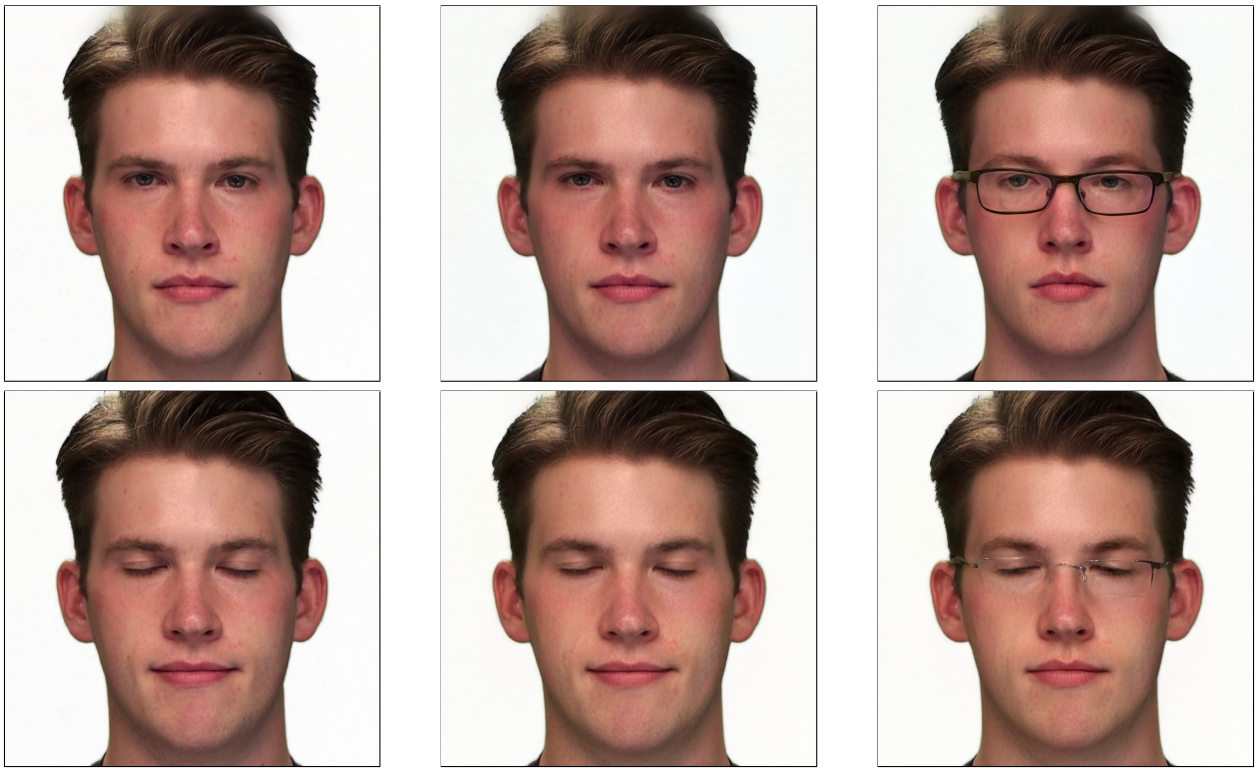
Generative adversarial models (GANs) [25] have shown remarkable ability to generate photorealistic images in various domains such as faces and scenes [90, 121, 122]. GANs take a latent code (usually sampled from a Gaussian distribution) as input and produce an image as the output. *GAN inversion* techniques allow us to project a *real image* onto the latent space of a pretrained GAN and retrieve its corresponding latent code. The pretrained GAN generator can then reconstruct that image using the estimated latent code. Modifying this estimated latent code opens up exciting new opportunities to perform a wide range of high-level editing tasks that are traditionally challenging, e.g., changing semantic object classes, modifying high-level attributes of the object/scene, or even applying 3D geometric transformations. We refer to the modification of the latent code with a semantic change in the image as *semantic editing*, e.g., changing the semantic attributes of an object.

Semantic editing in images. A recent line of research work [11, 12, 123, 124, 125, 126, 127] has shown promising results in reconstructing an input image by either optimizing the latent code (or latent variables) or directly predicting the latent code via an image encoder. These GAN inversion techniques enable interesting semantic photo editing applications. For image-level editing applications, several approaches [128, 129, 130] find specific semantic

directions in the latent space, e.g., changing poses, colors, or age, while others [131] aim to change the global style, e.g., photo → sketch, photo → cartoon. We denote them as *In-domain* and *Out-of-domain* editing, respectively. With these *image* GAN inversion-based semantic editing approaches, how can we extend them to *videos*?

Per-frame editing. One straightforward approach is to apply existing GAN inversion techniques [11, 12, 126, 127] for each frame in a video *independently*. Figure 4.2 shows an example of applying a StyleCLIP mapper [10] on two frames. The input and the independently reconstructed frames look plausible when viewed individually. However, the two edited frames exhibit inconsistency (e.g., the frame of the eyeglasses). Very recently, Yao et al. [15] learns to predict per-frame semantic editing directions for editing face videos. However, the edited videos suffer from apparent temporal flickering and fail to preserve facial identity.

Our work. In this paper, we present a method for *temporally consistent* video semantic editing. We start from the existing GAN inversion approaches [11, 12] to obtain the latent code for each frame. We first modify the latent code to achieve the initial per-frame editing results. However, such a direct editing approach results in temporal inconsistencies in the modified video's appearance or style. To deal with this challenge, we propose to compute bi-directional optical flow estimated from a frame pair sampled from the video. We can then adjust the latent code and the generator to minimize the photometric loss (along with valid flow vectors). We present a two-phase optimization strategy. In the first phase, we update only the latent codes via an MLP (with generator parameters frozen) to adjust the consistency of the detailed appearance. In the second phase, we finetune the generator with a local regularization to maintain the editability of the latent space. Our two-phase optimization approach helps achieve significantly improved temporal consistency while preserving the edited contents.



Input Inverted Edited

Figure 4.2: **Issues with per-frame editing.** While current methods achieve faithful inversion and photorealistic editing, the results are inconsistent across frames (*eyeglasses*) and may fail to preserve details of the input video (*lips*).

Concurrent work. Two concurrent work [132, 133] also apply StyleGAN for video editing. These methods either use per-frame pivot tuning [11] for maintaining the similarity between the edited and input frame [132] or apply latent vector smoothing [133] with StyleGAN3 [134]. Our method differs in 1) the use of explicit temporal consistency optimization and 2) the applicability of performing both in-domain and out-of-domain editing.

Our contributions

- We tackle a task on GAN-based semantic editing in videos. We propose a simple yet effective flow-based approach to mitigate the temporal inconsistency of a directly (frame by frame) edited video.
- We present a two-phase optimization approach for updating the latent code *and* generator to preserve the video details.
- Our method is agnostic and can be applied to different GAN inversion and editing approaches.

4.2 Related Work

Generative adversarial networks. The quality and resolution of generated images have been achieved rapidly in recent years [90, 121, 122, 134, 135]. These GAN models can map a random latent code (a noise vector) to a photorealistic image. Many recent efforts have been devoted to improving the generator architectures [121, 122, 134, 136], training strategies [90], loss function designs [137, 138], and regularization [139]. Our work builds upon existing pretrained StyleGAN models as they demonstrate disentangled latent space for editing. Instead of *generating synthetic images*, our goal is to *edit real videos*.

GAN inversion. GAN inversion [123, 140] allows us to reconstruct real images by projecting them onto a pretrained GAN's latent space. These techniques facilitate interesting photo editing applications. They can be split into encoder-based [12, 97, 141, 142, 143, 144, 144, 145, 145, 146], optimization-based [99, 106, 124, 125, 126, 147, 148, 149], and hybrid methods [11, 127, 150]. Our method is *agnostic* to different GAN inversion approaches for initializing the latent code. For example, our experiments explore using PTI inversion [11] for in-domain editing and Restyle encoder [12] for out-of-domain editing.

Semantic image editing in latent space. Semantic image manipulation and editing allow us to change the content and style of an image. It can be grouped into In-Domain editing and Out-of-Domain editing. Targeting at finding semantic directions in the latent space of a pretrained generator, in-domain editing [10, 98, 128, 129, 130, 151, 152, 153, 154, 155, 156, 157] manipulates the attributes of the object, but keeps the same style. Out-of-domain [131, 158, 159], however, aims to change the style of the image. These techniques usually perform well on a single image but fail to maintain temporal consistency if applied to a video.

Semantic video editing. Recent and concurrent work [15, 132, 133] explore *video editing* with a pre-trained StyleGAN. The methods in [15, 132] apply per-frame editing and show coherent editing without using any temporal information. However, these methods support only in-domain editing. For *localized editing* (e.g., adding eyeglasses), we find that the method in [15] produces inconsistency and fails to preserve identity. The work [133] applies temporal smoothing on the *inverted latent vectors* in StyleGAN3 [134]. Our approach, in contrast, directly minimizes the temporal photometric inconsistency at the *synthesized frames*.

Video editing and temporal consistency. Temporal consistency is one critical cri-

terion in video editing. Existing methods achieve temporal consistency often by enforcing the output videos to satisfy the constraints imposed by 2D optical flow [160, 161]. Alternatively, several methods first estimate an unwarped 2D texture map (either explicitly [162] or implicitly [163]) and then perform editing, e.g., adding a pattern or changing the style of the 2D unwarped textures. The editing can then be propagated to the original video via the estimated UV mapping. Several *blind* methods enhance the temporal consistency as a *post-processing* step [16, 51, 164]. However, they typically have difficulty in handling videos with significant appearance changes. Our work shares similar ideas with these methods to enforce temporal consistency, using the optical flow fields estimated from the initial edited video. Instead of directly optimizing the *pixel values*, our core idea is to leverage the pre-trained generator, update the latent code and generator to achieve temporal consistent *and* photorealistic results.

4.3 Method

4.3.1 Overview

GAN Inversion. Given an input video $V_{input} = \{I_1, \cdots, I_T\}$ of T frames, our goal is to semantically edit all the video frames while preserving the temporal coherence of the edited video. To edit the input video V_{input}, we first align its frames by using a facial alignment method [165]. Then we use existing GAN inversion techniques (e.g., [11, 12]) to invert the frames back to the latent code such that the inverted frame $I_t^{inv} = G(W_t^{inv}; \theta^{inv})$ is similar to the input frame: $I_t^{inv} \approx I_t$. With the inverted frames, we can edit the inverted video $V_{inv} = \{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ by *independently* editing its frames I_t^{inv}. We denote this frame-by-frame editing approach as "direct editing".

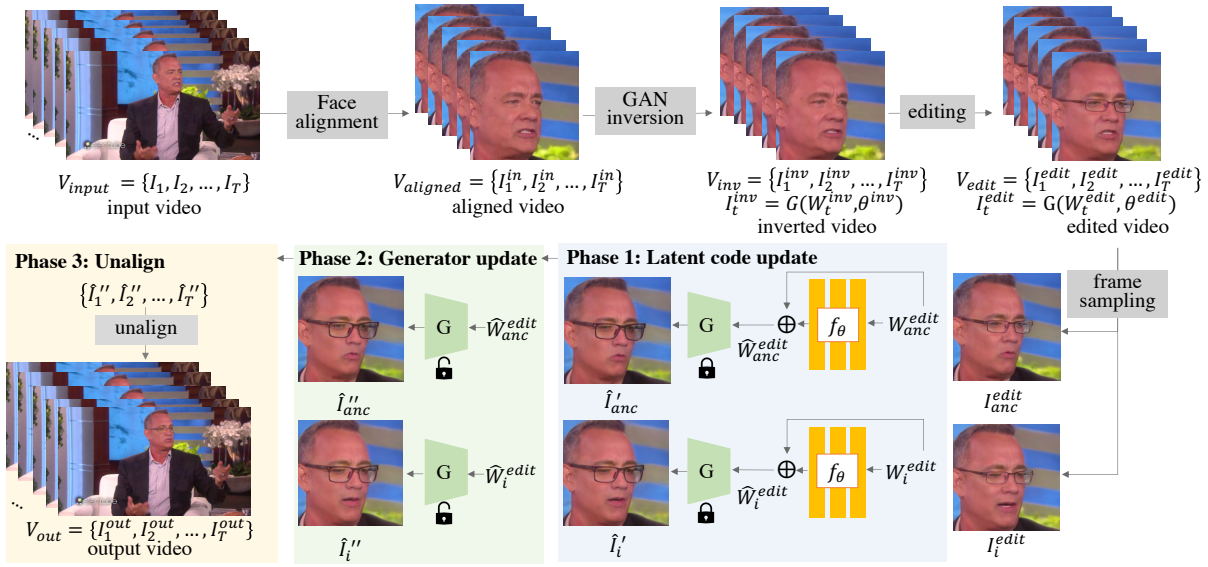
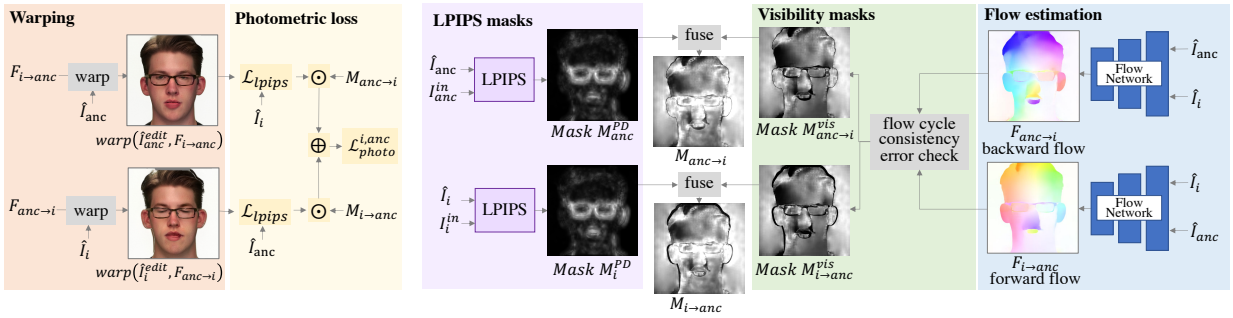


Figure 4.3: **Video editing with flow-based temporal consistency.** Given an input video of T frames V_{input}, we first spatially align the video frames using an off-the-shelf face landmark detector. We then use existing GAN inversion techniques [11, 12] to obtain the inverted frames $\{I_1^{inv}, I_2^{inv}, \cdots, I_T^{inv}\}$ and their corresponding latent code in the $\mathcal{W}^+$-space of StyleGAN $\{W_1^{inv}, W_2^{inv}, \cdots, W_T^{inv}\}$. We independently perform semantic editing on these inverted frames to obtain $\{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$ and their corresponding latent code $\{W_1^{edit}, W_2^{edit}, \cdots, W_T^{edit}\}$. To achieve temporal consistency, we choose an anchor frame I_{anc}^{edit} as the reference frame, and each time sample another frame I_i^{edit} from the edited video. To generate a temporally consistent edited video, we first refine the latent codes of the directly edited video W_{anc}^{edit} and $\{W_i^{edit}\}_{i \neq anc}$ to $\hat{W}_{anc}^{edit}$ and $\{\hat{W}_i^{edit}\}_{i \neq anc}$ by optimizing an MLP f_θ (phase 1). These refined latent codes result in the temporally consistent frames $\hat{I}_{anc}'$ and $\hat{I}_i'$. To further improve the temporal consistency, we keep the refined latent codes $\hat{W}_{anc}^{edit}$ and W_i^{edit} and only update the generator parameters (phase 2). This will generate $\hat{I}_{anc}''$ and $\hat{I}_i''$ with improved temporal consistency. After our two phase optimization, we finally unalign the frames to generate our final edited video V_{out} (phase 3).



(a) $\mathcal{L}_{photo}$ computation (b) Flow and mask computation

Figure 4.4: **Photometric loss for temporal consistency.** Given a frame pair $\hat{I}_i$ and $\hat{I}_{anc}$ (either from phase 1 or phase 2), we compute the forward and backward flows $F_{i\to anc}$ and $F_{anc\to i}$ using RAFT [13]. We then use these two flow fields to compute the visibility masks by performing a forward-backward and backward-forward flow consistency error check. For in-domain editing, we also use LPIPS to obtain a semantic mask that highlights the difference between the aligned input frames I_i^{in} and I_{anc}^{in} and our edited frames $\hat{I}_i$ and $\hat{I}_{anc}$. We then fuse both the LPIPS semantic masks and the visibility masks to get our final masks $M_{anc\to i}$ and $M_{i\to anc}$. To compute the photometric loss (Eqn. 4.1), we use the flows to warp the directly edited frames and utilize the fuzed masks as shown in (a).

In-domain and out-of-domain GAN-based editing. Commonly used *image-based* editing techniques via a GAN include (1) in-domain and (2) out-of-domain editing. We refer to an *in-domain editing* [128, 129, 130, 151] as the editing that only manipulates the latent code, given a *fixed* pretrained generator. That is, the generator parameters θ^{inv} remain frozen ($\theta^{inv} = \theta^{edit}$), and only the latent code W_t^{edit} is updated. The in-domain editing usually changes semantic attributes such as color, age, or facial expressions. On the other hand, out-of-domain editing may involve updating the pretrained generator to produce an entirely new style (as shown in [131]). Here, the latent code remains the same $W_t^{edit} = W_t^{inv}$ and only the generator θ^{edit} changes.

Direct editing on a video. When applying both types of editing techniques to a video independently for each frame, we obtain an edited video $V_{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$. For each directly edited frame I_t^{edit}, there is a corresponding latent code W_t^{edit} such that $I_t^{edit} = G(W_t^{edit}; \theta^{edit})$. Due to the per-frame, independent process, the edited video V_{edit}

often suffers from temporal inconsistency. Moreover, due to the poor disentanglement of this per-frame editing, not only will the edited attributes differ among frames, but other existing facial attributes also change (see the change in mouth in Figure 4.5). Our goal is to ensure that the edited attributes remain temporally consistent while preserving the other details from the input video.

Overview of our approach. To achieve this goal, we propose a two-phase optimization approach: phase 1 updates the *latent code* via an MLP and phase 2 updates the *generator*. In both phases, we optimize the temporal photometric loss across frames. With the finetuned latent code and generator, we unalign the edited frames to produce an edited video. Figure 4.3 outlines our workflow. Below, we describe the details and the losses of our approach.

4.3.2 Flow-based temporal consistency

We present a flow-based approach to explicitly encourage temporal consistency in the edited video V_{edit}.

Frame sampling. As we cannot fit an entire video into the GPU memory, we choose to perform our optimization from a *pair of frames* at a time. We choose to use an anchor frame I^{edit}_{anc} as one of the pair, which we set as the middle frame of the video. This is inspired by recent video representation work [166], where a video is represented by a key frame and a flow network. At each iteration, we sample a latent code W^{edit}_i, corresponding to the frame I^{edit}_i and optimize the pair of frames $\{I^{edit}_{anc}, I^{edit}_i\}$. We perform our optimization in two phases (Section 4.3.3). In phase 1, we generate temporally consistent pairs $\{\hat{I}'_{anc}, \hat{I}'_i\}_{i \neq anc}$ as a result. In phase 2, we further improve the temporal consistency, recover other affected attributes brought by the per-frame editing due to the poor disentanglement, and generate the pairs

$\{\hat{I}''_{anc}, \hat{I}''_i\}_{i \neq anc}$.

Flow estimation and warping. We use RAFT [13] to compute the forward and backward flows $F_{i \to anc}$ and $F_{anc \to i}$ of the pair $\{\hat{I}_{anc}, \hat{I}_i\}$. This pair is either the output of phase 1 $\{\hat{I}'_{anc}, \hat{I}'_i\}$ or phase 2 $\{\hat{I}''_{anc}, \hat{I}''_i\}$. We then use these two flows to warp the pair of frames $\{\hat{I}_{anc}, \hat{I}_i\}$.

Visibility masks. To highlight the *non-occluded* regions, we compute the visibility masks $M^{vis}_{anc \to i}$ and $M^{vis}_{i \to anc} \in [0, 1]$. This mask shows lower weights for occluded pixels and higher weights for the non-occluded pixels (Figure 4.4). To compute the visibility masks, we first compute forward-backward and backward-forward flow consistency error maps $\epsilon_{anc \to i}$ and $\epsilon_{i \to anc}$ and compute the error map by $\epsilon_{i \to anc}(p) = ||p - F_{anc \to i}(p + F_{anc \to j}(p))||_2$, where p is a pixel in the flow field. These resultant error maps are mapped to $[0, 1]$ using an exponential function such that $M^{vis}_{anc \to i} = \exp(-10\epsilon_{anc \to i})$ and $M^{vis}_{i \to anc} = \exp(-10\epsilon_{i \to anc})$.

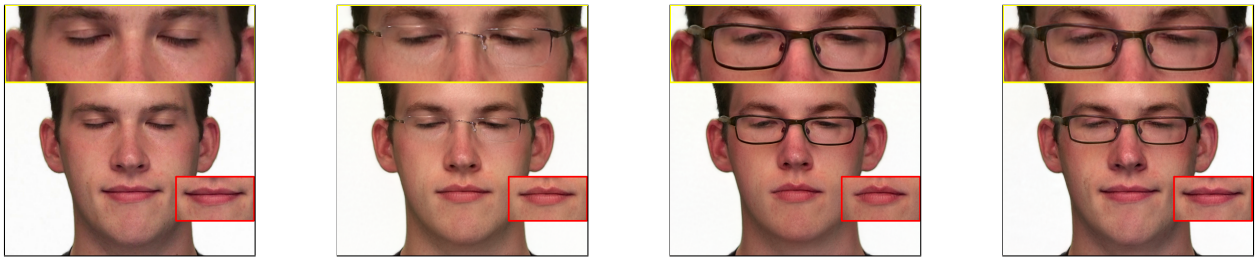
Perceptual difference mask. For in-domain editing, because the introduced editing is temporally inconsistent, we observe that the visibility masks do *not* emphasize those edited parts (e.g., eyeglasses). To highlight those edited parts, we compute the soft semantic perceptual difference masks M^{PD}_{anc} and M^{PD}_i between the pair of frames and their corresponding aligned input frames using LPIPS [167] (Figure 4.4). Due to the significant appearance differences, we cannot use these semantic perceptual difference masks for out-of-domain editing.

Fused masks. For in-domain editing, we fuse the visibility masks and the semantic perceptual difference masks such that $M_{anc \to i} = (M^{vis}_{anc \to i} \oplus M^{PD}_i)$ and $M_{i \to anc} = (M^{vis}_{i \to anc} \oplus M^{PD}_{anc})$. The masks will also be clamped to [0, 1]. This fusion is shown in Figure 4.4. On the other hand, for out-of-domain editing, $M_{anc \to i} = M^{vis}_{anc \to i}$ and $M_{i \to anc} = M^{vis}_{i \to anc}$.

Bi-directional photometric loss. We use the warped frames and the final computed masks to compute the bi-directional photometric loss to achieve a temporally consistent video. This loss measures the difference between the two frames to calculate the deviation in the non-occluded parts.

$$\begin{aligned}\mathcal{L}_{photo} = \sum_{\hat{I}_i, \hat{I}_{anc} \in P} & M_{i \to anc} \mathcal{L}_{LPIPS}(\hat{I}_{anc}, warp(\hat{I}_i, F_{anc \to i})) \\ & + M_{anc \to i} \mathcal{L}_{LPIPS}(\hat{I}_i, warp(\hat{I}_{anc}, F_{i \to anc}))\,,\end{aligned} \tag{4.1}$$

where $\hat{I}_t$ is either the output of phase 1 $\hat{I}'_t$ or phase 2 $\hat{I}''_t$. Intuitively, this bi-directional photometric loss ensures colors along the valid (forward-backward or backward-forward consistent) vectors across frames are as similar as possible.



Input | Direct Editing | W updated | W, G updated

Figure 4.5: **Motivation for two-phase optimization.** Updating latent code W brings in the eyeglasses, and tuning G with the perceptual difference mask recovers the expression in the input.

4.3.3 Two-phase optimization strategy

We split our optimization into two phases. In the first phase, we refine the latent codes $\{W_t^{edit}\}$ by only optimizing an MLP f_θ. While in the second phase, we only update the generator weights θ^{edit}.

Motivation. We use a two-phase optimization approach for in-domain editing because we observe that only refining the latent codes (phase 1) sometimes introduces undesired changes to *other* facial attributes. We show an example in Figure 4.5. When we only update the latent codes, we achieve temporal consistency of the introduced glasses; however, the mouth expression of the person changes. To address this in the case of in-domain editing, we update the generator weights (phase 2) using the perceptual difference mask to enforce the pixels outside the mask to be the same as the input. This will maintain the facial expression of the aligned input frame. The primary source of inconsistency for out-of-domain editing is the global inconsistency (e.g., background). Hence, updating the generator (phase 2) introduces this desired global change.

Phase 1: Latent code update. In this phase, we update the latent code W_t^{edit} using a Multi-layer Perceptron (MLP) $f_\theta = (w; \theta_f)$ implicitly. We use the same architecture as StyleCLIP mapper [10]. We use this MLP to predict a residual for the latent codes and update the parameters of the MLP instead of directly optimizing the latent codes explicitly, such that:

$$\hat{W}_t^{edit} = W_t^{edit} + \alpha f_\theta(W_t^{edit}; \theta_f)\,, \tag{4.2}$$

then for a pair of directly edited frames $\{I_{anc}^{edit}, I_i^{edit}\}$, we can get the updated frames $\hat{I}_i' = G(\hat{W}_i^{edit})$, $\hat{I}_{anc}' = G(\hat{W}_{anc}^{edit})$.

Our goal is to minimize:

$$\underset{\theta_f}{\operatorname{argmin}}\, \mathcal{L}_I = \underset{\theta_f}{\operatorname{argmin}} \sum_{t \neq anc} \mathcal{L}_{photo} + \lambda_{rf}\mathcal{L}_{rf} + \lambda_\epsilon \mathcal{L}_\epsilon\,, \tag{4.3}$$

where $\mathcal{L}_{photo}$ is the photometric loss, and

$$\mathcal{L}_{rf} = ||f_\theta(W_t^{edit}; \theta_f)||_1 + ||f_\theta(W_{anc}^{edit}; \theta_f)||_1 \tag{4.4}$$

is a regularization term to make sure we do not deviate too much from W_t^{edit}. We set $\lambda_{rf} = 0.1$ for the experiments. $\mathcal{L}_\epsilon = ||\epsilon_{anc\to i}||_1 + ||\epsilon_{i\to anc}||_1$ is the norm of error maps, and we set $\lambda_\epsilon = 10$.

The reason we use an MLP to update the latent code *implicitly* is that we observe that *explicitly* optimizing the latent codes results in an unstable optimization when using a large learning rate. However, the running time becomes too long when using a small learning rate. To address this, we introduce an MLP to predict the residual and update the latent codes *implicitly*. This leads to a more stable optimization. We show an example of x-t scanline in Figure 4.6 to demonstrate the effectiveness of introducing the MLP.

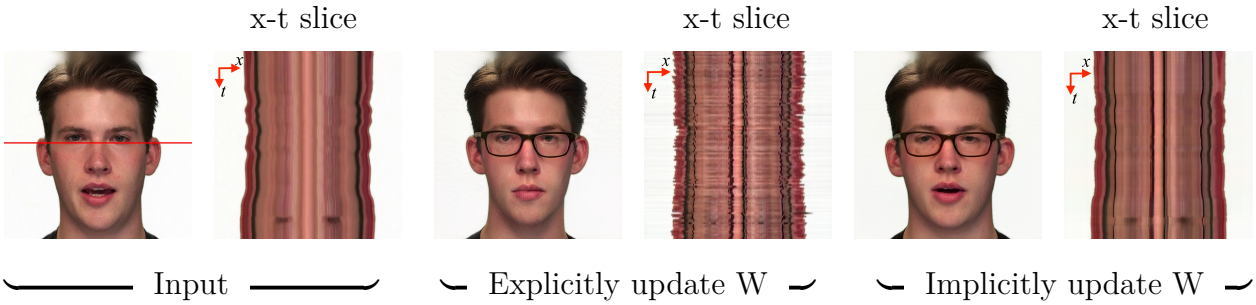


Figure 4.6: **x-t slices between updating latent codes explicitly and implicitly with an MLP.** We visualize the optimized frames and an x-t slice at $y = 500$. Explicitly updating latent code W gives us an unstable x-t scanline, while updating W implicitly with an MLP gives a smooth scanline.

Phase 2: Generator update. For in-domain editing, in this phase, we use the updated latent codes $\{\hat{W}_t^{edit}\}_{t=1}^T$ from phase 1, and our goal is to finetune the generator only to minimize:

$$\hat{\theta}^{edit} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}}\, \mathcal{L}_{II} = \underset{\hat{\theta}^{edit}}{\operatorname{argmin}} \sum_{t\neq anc} \mathcal{L}_{photo} + \lambda_\epsilon \mathcal{L}_\epsilon + \lambda_r \mathcal{L}_r + \lambda_M \mathcal{L}_M\,, \tag{4.5}$$

$$\mathcal{L}_M = (1 - M_i^{PD})\mathcal{L}_{LPIPS}(\hat{I}_i'', I_i^{in}) + (1 - M_{anc}^{PD})\mathcal{L}_{LPIPS}(\hat{I}_{anc}'', I_{anc}^{in})\,. \tag{4.6}$$

M_i^{PD} is the perceptual difference mask computed between $\hat{I}_i'' = G(\hat{W}_t^{edit}; \hat{\theta}^{edit})$ and aligned input I_i^{in}, and $\mathcal{L}_{LPIPS}(\cdot, \cdot)$ is the LPIPS distance loss [167]. We initialize $\hat{\theta}^{edit}$ as θ^{edit}. The LPIPS term also plays a role to maintain the sharpness of the edited frames. This is because the consistency can be achieved by pushing all the frames to become blurry.

Here, $\mathcal{L}_r$ is the regularization loss for the generator and λ_r is the strength of regularization. We introduce this loss to help prevent the generator G from losing its latent space editability as we do not wish to *ruin* its pretrained latent space. Therefore, similar to [11], we use this *local regularization* to preserve the editing ability of our generator. More specifically, we first obtain a latent code W_r by linearly interpolating between the current latent code $\hat{W}_t^{edit}$ and a randomly sampled code W_z with an interpolation parameter α_{interp}: $W_r = \hat{W}_t^{edit} + \alpha_{interp} \frac{W_z - \hat{W}_t^{edit}}{||W_z - \hat{W}_t^{edit}||_2}$. This gives us a new latent code in a local region around $\hat{W}_t^{edit}$.

To ensure that we do not lose the editing capability of the original generator, we add a penalty on the distance between the generated image from the new generator and the old one such that:

$$\mathcal{L}_r = \mathcal{L}_{LPIPS}(x_r, \hat{x}_r) + \lambda_{\ell_2}^r \mathcal{L}_{\ell_2}(x_r, \hat{x}_r)\,, \tag{4.7}$$

where $x_r = G(W_r; \theta^{edit}), \hat{x}_r = G(W_r; \hat{\theta}^{edit})$, $\lambda_{\ell_2}^r$ is the weight for ℓ_2 loss. This regularization can alleviate the side effects from updating G within a local area. This is desirable since for a video, the latent codes for the same identity tend to gather locally.

For out-of-domain editing, unlike in-domain editing, we cannot rely on the perceptual difference mask, so the optimization goal reduces to:

$$\hat{\theta}^{edit} = \underset{\hat{\theta}^{edit}}{\text{argmin}}\, \mathcal{L}_{II} = \underset{\hat{\theta}^{edit}}{\text{argmin}} \sum_{t \neq anc} \mathcal{L}_{photo} + \lambda_r \mathcal{L}_r + \lambda_\epsilon \mathcal{L}_\epsilon\,. \tag{4.8}$$

To compensate for the regularization effect of the perceptual difference mask, we freeze the

last eight layers of the synthesis network in G to avoid blurry results. As all the computations, including the GAN generator, flow estimation network, spatial warping, and photometric losses, are *differentiable*, we can backpropagate the errors all the way back. After phase 1 and 2, we will have $\{\hat{W}_t^{edit}\}_{t=1}^{T}$ and $G(\cdot;\hat{\theta}^{edit})$ as a result.

4.3.4 Phase 3: Unalign

After our two-phase optimization, we perform our final phase as post-processing. In this phase, we put the aligned frames back to the original video to generate our final edited video (see Figure 4.3). We follow the *stitch tuning* approach in [132] by tuning the generator to reduce the edge artifact brought by editing. Note that this is only feasible for the in-domain editing because the out-of-domain editing has a global appearance compared to the input video.

4.4 Experimental Results

4.4.1 Experimental setup

Implementation details. We use StyleGAN-ADA [168] as our pre-trained generator. We experiment with in-domain and out-of-domain editing techniques to validate our approach for different GAN inversion methods. Specifically, for in-domain editing, we use the PTI inversion [11] (based on e4e [145]) and StyleCLIP mapper [10]. For out-of-domain editing, we use the Restyle encoder [12] and the StyleGAN-NADA [131]. We will release the source code and pretrained models. In the following, we show sample results from the video frames. We encourage the readers to view the videos in the supplementary material for video results.

Datasets. We conduct our metric evaluation using 20 videos randomly sampled from RAVDESS dataset [14]. We conduct 5 types of in-domain editing for each video and 5 types of out-of-domain editing. To further demonstrate the capabilities of our method to handle *real* videos, we also apply our approach to Internet videos and show the visual results.

Metrics. We aim to evaluate the method using two main aspects: 1) temporal consistency and 2) perceptual similarity with the semantically edited frames. To evaluate temporal consistency, we measure the *Warping Error* E_{warp}:

$$E_{warp}(I_t, I_{t+1}) = \frac{1}{\sum_{i=1}^{N} M_t(p_i)} \cdot \sum_{i=1}^{N} M_t(p_i) ||I_t(p_i) - \hat{I}_{t+1}(p_i)||_2^2 , \tag{4.9}$$

where $\hat{I}_{t+1} = warp(I_{t+1}, F_{t \to t+1})$, N is the number of pixels, p_i is the i-th pixel, M_t is a binary non-occlusion mask, which shows non-occluded pixels, we compute it using the forward-backward consistency error the threshold in [169, 170].

We also measure the LPIPS perceptual similarity score [167] (with AlexNet [171]) between the directly edited video $V^{edit} = \{I_1^{edit}, I_2^{edit}, \cdots, I_T^{edit}\}$ and the output of our phase 2 $\{\hat{I}_1'', \hat{I}_2'', \cdots, \hat{I}_T''\}$ by measuring the averaged perceptual similarity between the corresponding frames. The purpose of these two metrics is to evaluate whether the method can achieve a balance between *temporal consistency* and *fidelity degradation*. This is an inherent trade-off. Preserving all the details of per-frame editing inevitably leads to temporal flickering artifacts. Focusing only on temporal consistency may easily lead to blurry videos. Our goal is that the final output video is visually similar to the directly (per-frame) edited video. Our goal is that the final output video is visually similar to the directly (per-frame) edited video.

Table 4.1: **Out-of-domain editing comparison.**

	$E_{warp} \downarrow$		LPIPS↓	
Direct editing	0.0098		0.0000	
Editing categories	DVP [16]	Ours	DVP	Ours
Sketch	0.0036	0.0085	0.2404	0.1314
Pixar	0.0031	0.0025	0.1074	0.1178
Disney Princess	0.0040	0.0078	0.2062	0.1204
Elf	0.0042	0.0108	0.2289	0.1310
Zombie	0.0040	0.0085	0.2033	0.1370
Average perfomance	0.0038	0.0076	0.1972	0.1275

4.4.2 Out-of-domain results

Setup. We first invert the videos frame by frame using the Restyle encoder [12] (psp-based [144]). We then directly apply five different out-of-domain editing effects produced by StyleGAN-NADA [131]. We perform our two-phase optimization approach on the directly edited video using Adam optimizer [172]. For phase 1, we set the learning rate to $\alpha_I = 0.005$, and update the latent codes for 5 epochs. In Eqn. 4.2, we set $\alpha = 0.04$ for all the editing directions. For phase 2, we set the learning rate to $\alpha_{II} = 8 \times 10^{-4}$, and finetune G for 5 epochs. We set the regularization weight λ_r to 200.

Evaluation. Table 4.1 shows that our method decreases the temporal error of the directly edited video. The primary sources of inconsistency in out-of-domain editing can be seen in the flickering background and the details of the hair. We show our visual results in Figure 4.7. Our method preserves the temporal consistency and maintains the sharpness of the input video.

4.4.3 In-domain editing results

Setup. We first invert the videos frame by frame by using the PTI method [11]. We then directly apply five different semantic editing directions discovered by StyleCLIP mapper [10].

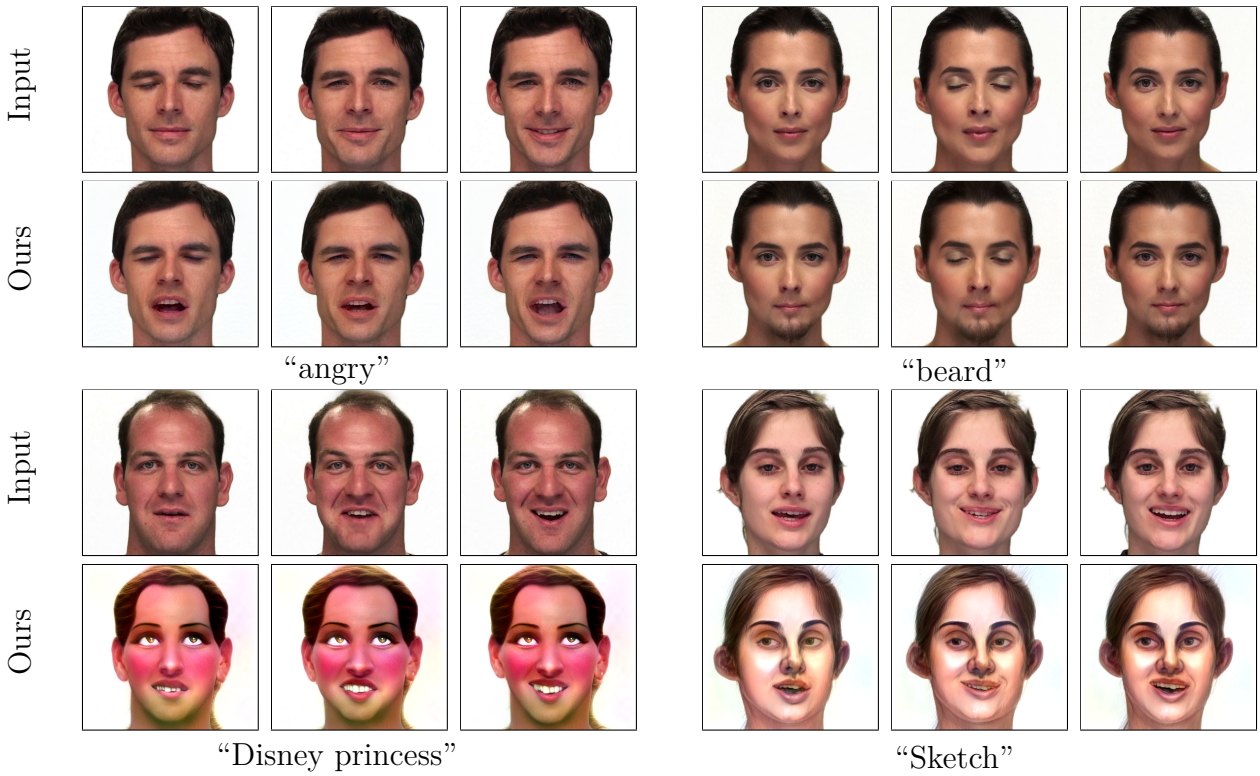


Figure 4.7: **Visual results on RAVDESS dataset [14].** We show both in-domain ("angry" and "eyeglasses") and out-of-domain ("Disney princess" and "Sketch") editing results. Our results maintain consistent changes with time preserving the temporal coherence.

Table 4.2: **In-domain editing comparison.**

	$E_{warp} \downarrow$		LPIPS↓	
Direct editing	0.0076		0.0000	
Editing categories	DVP [16]	Ours	DVP	Ours
angry	0.0033	0.0032	0.2452	0.1100
beard	0.0038	0.0030	0.2444	0.1033
eyeglasses	0.0039	0.0034	0.1226	0.1097
Depp	0.0037	0.0031	0.2452	0.2024
surprised	0.0035	0.0028	0.1415	0.1012
Average perfomance	0.0036	0.0031	0.1760	0.1253

Next, we perform our two-phase optimization approach on the directly edited video using Adam optimizer [172]. For phase 1, we set the learning rate $\alpha_I = 0.05$, and update f_θ for 10 epochs. In Eqn. 4.2, we set $\alpha = 0.12$ for the "eyeglasses", and $\alpha = 0.04$ for the rest of the semantic directions. For phase 2, we set the learning rate of G to $\alpha_{II} = 0.0001$, and finetune G for 5 epochs. We set the regularization weight λ_r to 200.

Evaluation. Table 4.2 shows that our approach improves the temporal consistency over the directly edited video baseline by a large margin. When dealing with in-domain editing, the primary source of inconsistency is the details of the newly added attributes, e.g., glasses or beard and some background flickering. We show sample visual results in Figure 4.7, where the introduced changes are consistent among the different frames.

4.4.4 Two-phase optimization strategy ablation study

We demonstrate the effect of our two-phase optimization strategy of updating the latent codes first and following that with finetuning the generator G. We compare our two-phase approach to: (1) No optimization (i.e., direct editing), (2) update latent code only (phase 1), and (3) finetune generator G only. We show the results in Table 4.3. When we only update generator G, we can achieve a low warping error E_{warp}. However, this is not desirable since finetuning G pushes the video to be consistent globally without modifying the local details.

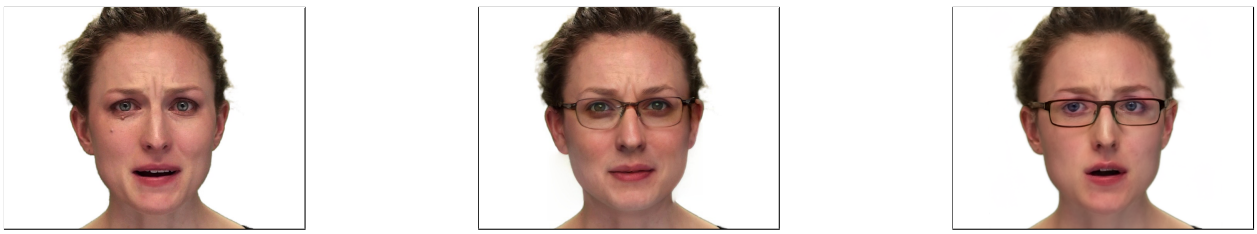
Table 4.3: **Two-stage optimization strategy ablation study.**

Optimization stage		In-domain editing		Out-of-domain editing	
Update W_t^{edit}	Update G	$E_{warp} \downarrow$	LPIPS↓	$E_{warp} \downarrow$	LPIPS↓
-	-	0.0076	0.0000	0.0098	0.0000
✓	-	0.0064	0.2108	0.0094	0.1428
-	✓	**0.0027**	0.2463	**0.0057**	0.1375
✓	✓	0.0031	**0.1253**	0.0076	**0.1275**

Therefore, the output video is different from the directly edited video (i.e., high LPIPS distance). Thus, we follow our two-phase optimization of a) updating the latent codes via an MLP f_θ (to improve local consistency), b) finetuning the generator G (to modify the global effect).

4.4.5 Comparison with Latent Transformer

We compare our method with Latent Transformer (LT) [15]. We show a qualitative comparison in Fig. 4.8. LT edits video by updating the projected latent code *independently* for each frame without using temporal constraints. Our method, in contrast, uses flow-based loss to improve the temporal consistency, and our second phase uses a perceptual difference mask as a regularization to preserve the facial details other than the edited parts. As a result, our method can improve temporal consistency and preserve personal identity.



Input LT Ours

Figure 4.8: **Visual comparison with Latent Transformer (LT) [15].** LT cannot preserve the person's identity very well. Our method can preserve the identity and achieves a temporal consistent video.

4.4.6 Comparison with Deep Video Prior (DVP)

We compare our method with DVP [16], a state-of-the-art blind video consistency approach, using their default setting. We show the in-domain editing comparison in Table 4.2 and the out-of-domain editing comparison in Table 4.1. For warping error E_{warp}, our method achieves improved results for in-domain editing and comparable results for out-of-domain editing. However, in terms of LPIPS distance, our visual results are more similar to the directly edited video for both in-domain and out-of-domain editing. We show visual comparison in Figure 4.9. DVP can achieve temporally consistent results (i.e., low E_{warp}). However, this is at the cost of losing local details in the "eyeglasses" example or excessively smoothing the results to get a blurry video as in the "Disney Princess" example.

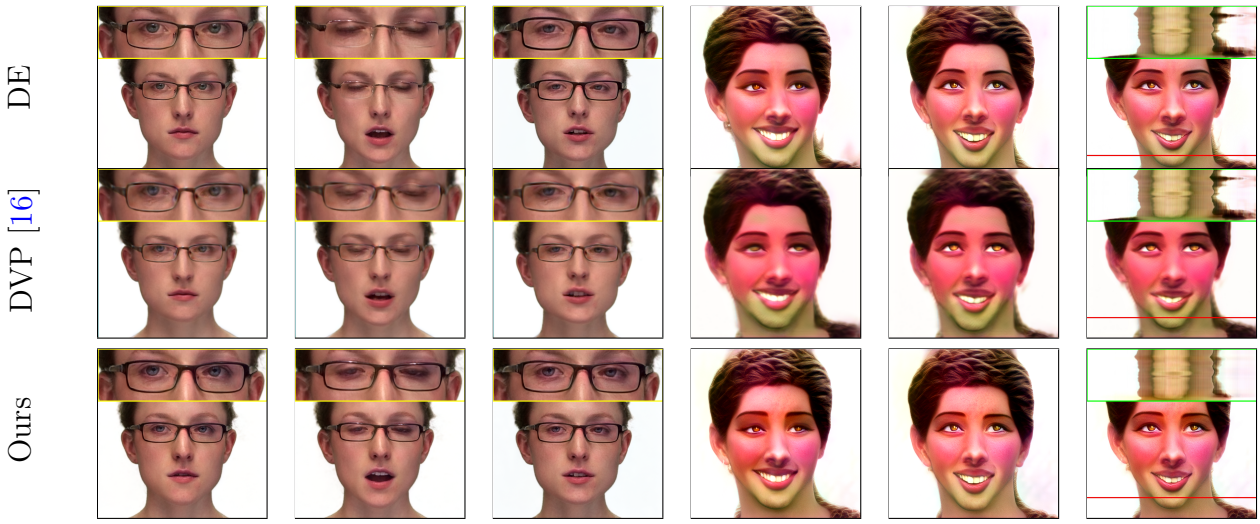


Figure 4.9: **Visual comparison with DVP [16].** DVP achieves temporal consistency by severely smoothing the image and hence losing its sharpness. Our method, however, can achieve a balance between consistency and sharpness. In "eyeglasses" example (left), DVP shows a different pair of eyeglasses across the time (zoom-in for better visualization), while ours remain a good consistency for the eyeglasses; in "Disney princess" (right), DVP shows a blurry result with an unstable x-t scanline, while ours is sharper and shows a stable consistency in the scanline.

4.4.7 In-the-wild results

To demonstrate the capabilities of our method to handle *real* videos, we apply our approach to Internet videos and show the visual results in Figure 4.10.

Figure 4.10: **Results on Internet videos**. Results on the Internet videos. We change the first person to "surprised" expression, and change the second person to "angry".

4.4.8 Limitations

We show several limitations of our approach in Figure 4.11. First, our approach relies on plausible results from existing GAN inversion and editing techniques. We show an example of added earrings in Figure 4.11(a), and an example of a rare pose in Figure 4.11(a). Second, the GANs used in our experiments require the objects to be spatially aligned and thus may not yet be suitable for inverting and editing unconstrained videos. Third, our method relies on a high-quality GAN model that may be computationally expensive to train and often

require diverse training images. Our full method (phases 1, 2 and 3) takes 40 minutes on a 150-frame video, on a single NVIDIA P6000 GPU.

(a) "Zombie" (b) "Rare pose"

Figure 4.11: **Limitations**. From (a), it can be seen that earrings are added by GAN editing prior to our flow-based temporal consistency approach. Since our approach builds on existing GAN inversion and editing techniques, it will be affected by their quality. From (b), it can be seen that our method fails when there is a rare pose and a large motion.

4.5 Conclusions

We have presented a novel method for video-based semantic editing by leveraging image-based GAN inversion and editing. We first perform direct per-frame editing, and then refine the results using a flow-based method to minimize the bi-directional photometric loss. To achieve temporal consistency, we adjust the latent codes via a multi-layer perceptron (MLP) and fine-tune the generator (G). We show that the resulting edited video maintains its similarity to the direct editing results and exhibits temporal consistency. Furthermore, our method is model-agnostic, allowing for its application with different GAN inversion and manipulation techniques.

Chapter 5

Conclusion

In this work, we tackled several controllable image synthesis problems. In Chapter 2, we first proposed a generic approach for guided image-to-image translation. Although the proposed approach can be applied to different guided image-to-image tasks, the conditioning scheme builds on top of CNN decoders, which have limitations in synthesizing high-resolution detailed content.

Therefore, in Chapter 3 we overcome this by extending the StyleGAN generator so that it takes pose as input for controlling poses and introduce a spatially varying modulation for the latent space for controlling appearances. This enables us to realistically hallucinate occluded content while preserving the identity and fine details of the source image. However, the resulting synthesized image is limited to 2D space and is not temporally consistent nor view consistent.

Thus we tackle temporal consistent synthesis in Chapter 4. We propose an approach to edit video frames using GAN-based editing techniques and flow-based refinement.

Future work. Our proposed methods have demonstrated promising results. However, there are still more problems to tackle in order to push the boundaries of what is possible. The following are some potential directions for future work:

- View consistent synthesis is an important research direction to explore in the future. While our current methods can achieve temporal consistency, it would be interesting

to achieve view consistency. This can have numerous applications such as in sports, where analysts can dive into the field and analyze the game from different angles.

- Capture the whole scene! Currently, our focus is on a single human as the subject of interest. However, including multiple subjects as well as the background and capturing entire scenes would be an interesting avenue to explore. This would enable a more immersive experience, allowing viewers to explore the scene from any novel viewpoint.

Bibliography

[1] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: Controlling deep image synthesis with texture patches," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[3] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[6] Y. Men, Y. Mao, Y. Jiang, W.-Y. Ma, and Z. Lian, "Controllable person image synthesis with attribute-decomposed GAN," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[7] Y. Ren, X. Yu, J. Chen, T. H. Li, and G. Li, "Deep image spatial transformation for person image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[8] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] K. Sarkar, V. Golyanik, L. Liu, and C. Theobalt, "Style and pose control for image synthesis of humans from a single monocular view," *arXiv preprint arXiv:2102.11263*, 2021.

[10] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of StyleGAN imagery," in *IEEE International Conference on Computer Vision*, pp. 2085–2094, October 2021.

[11] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, "Pivotal tuning for latent-based editing of real images," *ACM Transactions on Graphics*, 2021.

[12] Y. Alaluf, O. Patashnik, and D. Cohen-Or, "Restyle: A residual-based StyleGAN encoder via iterative refinement," in *IEEE International Conference on Computer Vision*, October 2021.

[13] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *European Conference on Computer Vision*, 2020.

[14] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS one*, vol. 13, pp. 1–35, 05 2018.

[15] X. Yao, A. Newson, Y. Gousseau, and P. Hellier, "A latent transformer for disentangled face editing in images and videos," in *IEEE International Conference on Computer Vision*, pp. 13789–13798, 2021.

[16] C. Lei, Y. Xing, and Q. Chen, "Blind video temporal consistency via deep video prior," in *Advances in Neural Information Processing Systems*, 2020.

[17] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Neural Information Processing Systems*, 2017.

[19] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*, 2012.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Association for the Advancement of Artificial Intelligence*, 2018.

[23] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *International Conference on Learning Representations*, 2017.

[24] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *IEEE International Conference on Computer Vision*, 2017.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Neural Information Processing Systems*, 2014.

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[27] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision*, 2017.

[29] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *IEEE International Conference on Computer Vision*, 2017.

[30] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Neural Information Processing Systems*, 2017.

[31] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *European Conference on Computer Vision*, 2018.

[32] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *European Conference on Computer Vision*, 2018.

[33] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Neural Information Processing Systems*, 2017.

[34] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[35] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, 2018.

[36] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "CrDoCo: Pixel-level domain transfer with cross-domain consistency," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[37] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Neural Information Processing Systems*, 2018.

[38] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics*, vol. 23, pp. 689–694, 2004.

[39] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 309–320, 2007.

[40] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics*, vol. 9, no. 4, 2017.

[41] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Transactions on Graphics*, vol. 34, no. 4, 2015.

[42] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[43] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," in *British Machine Vision Conference*, 2017.

[44] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *European Conference on Computer Vision*, 2018.

[45] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Joint image filtering with deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[46] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Neural Information Processing Systems*, 2017.

[47] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[48] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for pose-based human image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[49] N. Neverova, R. A. Güler, and I. Kokkinos, "Dense pose transfer," in *European Conference on Computer Vision*, 2018.

[50] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *IEEE International Conference on Computer Vision*, 2019.

[51] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *European Conference on Computer Vision*, 2018.

[52] J. Diebel and S. Thrun, "An application of Markov random fields to range sensing," in *Neural Information Processing Systems*, 2006.

[53] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[54] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *ACM Transactions on Graphics*, vol. 26, pp. 96–102, 2007.

[55] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[56] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *European Conference on Computer Vision*, 2016.

[57] J. T. Barron and B. Poole, "The fast bilateral solver," in *European Conference on Computer Vision*, 2016.

[58] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," in *European Conference on Computer Vision*, 2016.

[59] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised person image synthesis in arbitrary poses," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[60] P. Esser, E. Sutter, and B. Ommer, "A variational U-Net for conditional appearance and shape generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[61] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *IEEE International Conference on Computer Vision*, 2019.

[62] A. Grigorev, A. Sevastopolsky, A. Vakhitov, and V. Lempitsky, "Coordinate-based texture inpainting for pose-guided human image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[63] V. Lazova, E. Insafutdinov, and G. Pons-Moll, "360-degree textures of people in clothing from a single image," in *International Conference on 3D Vision*, 2019.

[64] K. Sarkar, D. Mehta, W. Xu, V. Golyanik, and C. Theobalt, "Neural re-rendering of humans from a single image," in *European Conference on Computer Vision*, 2020.

[65] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM Trans. Graphics*, vol. 34, no. 6, pp. 1–16, 2015.

[66] K. M. Lewis, S. Varadharajan, and I. Kemelmacher-Shlizerman, "Vogue: Try-on by StyleGAN interpolation optimization," *arXiv preprint arXiv:2101.02285*, 2021.

[67] K. Aberman, M. Shi, J. Liao, D. Lischinski, B. Chen, and D. Cohen-Or, "Deep video-based performance cloning," in *Computer Graphics Forum*, vol. 38, pp. 219–233, 2019.

[68] J. S. Yoon, L. Liu, V. Golyanik, K. Sarkar, H. S. Park, and C. Theobalt, "Pose-guided human animation from a single image in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[69] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[70] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor, "Tex2shape: Detailed full human body geometry from a single image," in *IEEE International Conference on Computer Vision*, 2019.

[71] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, "Neural rerendering in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[72] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[73] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graphics*, vol. 38, no. 4, pp. 1–12, 2019.

[74] A. Raj, J. Tanke, J. Hays, M. Vo, C. Stoll, and C. Lassner, "Anr: Articulated neural rendering for virtual avatars," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[75] L. Liu, W. Xu, M. Habermann, M. Zollhoefer, F. Bernard, H. Kim, W. Wang, and C. Theobalt, "Neural human video rendering by learning dynamic textures and

rendering-to-video translation," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[76] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, *et al.*, "State of the art on neural rendering," in *Computer Graphics Forum*, vol. 39, pp. 701–727, 2020.

[77] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *ACM Trans. Graphics*, vol. 38, pp. 65:1–65:14, July 2019.

[78] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision*, 2020.

[79] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R.-M. Brualla, "Nerfies: Deformable neural radiance fields," in *IEEE International Conference on Computer Vision*, 2021.

[80] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video," in *IEEE International Conference on Computer Vision*, 2021.

[81] W. Xian, J.-B. Huang, J. Kopf, and C. Kim, "Space-time neural irradiance fields for free-viewpoint video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[82] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view

synthesis of dynamic scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[83] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *IEEE International Conference on Computer Vision*, 2021.

[84] C. Gao, Y. Shih, W.-S. Lai, C.-K. Liang, and J.-B. Huang, "Portrait neural radiance fields from a single image," *arXiv preprint arXiv:2012.05903*, 2020.

[85] G. Gafni, J. Thies, M. Zollhöfer, and M. Nießner, "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[86] A. Noguchi, X. Sun, S. Lin, and T. Harada, "Neural articulated radiance field," in *IEEE International Conference on Computer Vision*, 2021.

[87] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, H. Bao, and X. Zhou, "Animatable neural radiance fields for human body modeling," in *IEEE International Conference on Computer Vision*, 2021.

[88] X. Zhang, S. Fanello, Y.-T. Tsai, T. Sun, T. Xue, R. Pandey, S. Orts-Escolano, P. Davidson, C. Rhemann, P. Debevec, *et al.*, "Neural light transport for relighting and view synthesis," *ACM Trans. Graphics*, vol. 40, no. 1, pp. 1–17, 2021.

[89] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International Conference on Machine Learning*, 2019.

[90] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

[91] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[92] A. Shoshan, N. Bhonker, I. Kviatkovsky, and G. Medioni, "GAN-Control: Explicitly controllable GANs," in *IEEE International Conference on Computer Vision*, 2021.

[93] W. Peebles, J. Peebles, J.-Y. Zhu, A. Efros, and A. Torralba, "The Hessian penalty: A weak prior for unsupervised disentanglement," in *European Conference on Computer Vision*, 2020.

[94] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANspace: Discovering interpretable GAN controls," in *Neural Information Processing Systems*, 2020.

[95] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[96] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," in *International Conference on Learning Representations*, 2020.

[97] A. Tewari, M. Elgharib, G. Bharaj, F. Bernard, H.-P. Seidel, P. Pérez, M. Zollhofer, and C. Theobalt, "Stylerig: Rigging StyleGAN for 3D control over portrait images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[98] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "Styleflow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows," *ACM Transactions on Graphics*, vol. 40, no. 3, pp. 1–21, 2021.

[99] A. Tewari, M. Elgharib, F. Bernard, H.-P. Seidel, P. Pérez, M. Zollhöfer, and C. Theobalt, "Pie: Portrait image embedding for semantic control," *ACM Trans. Graphics*, vol. 39, no. 6, pp. 1–14, 2020.

[100] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[101] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," in *Neural Information Processing Systems*, 2019.

[102] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *IEEE International Conference on Computer Vision*, 2019.

[103] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[104] Y. Alharbi and P. Wonka, "Disentangled image generation through structured noise injection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[105] L. Chai, J. Wulff, and P. Isola, "Using latent space regression to analyze and leverage compositionality in GANs," in *International Conference on Learning Representations*, 2021.

[106] E. Collins, R. Bala, B. Price, and S. Susstrunk, "Editing in style: Uncovering the local semantics of GANs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[107] A. Shocher, Y. Gandelsman, I. Mosseri, M. Yarom, M. Irani, W. T. Freeman, and T. Dekel, "Semantic pyramid for image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[108] H. Kim, Y. Choi, J. Kim, S. Yoo, and Y. Uh, "StyleMapGAN: Exploiting spatial dimensions of latent in GAN for real-time image editing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[109] B. AlBahar and J.-B. Huang, "Guided image-to-image translation with bi-directional feature transformation," in *IEEE International Conference on Computer Vision*, 2019.

[110] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised learning of probably symmetric deformable 3D objects from images in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[111] S. N. Sinha, K. Ramnath, and R. Szeliski, "Detecting and reconstructing 3D mirror symmetric objects," in *European Conference on Computer Vision*, 2012.

[112] R. A. Yeh, Y.-T. Hu, and A. G. Schwing, "Chirality nets for human pose regression," in *Neural Information Processing Systems*, 2019.

[113] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, "Instance-level human parsing via part grouping network," in *European Conference on Computer Vision*, 2018.

[114] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[115] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[116] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[117] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Neural Information Processing Systems*, 2017.

[118] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[119] B. Coors, A. P. Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *European Conference on Computer Vision*, 2018.

[120] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[121] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[122] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[123] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision*, 2016.

[124] R. Abdal, Y. Qin, and P. Wonka, "Image2styleGAN: How to embed images into the

StyleGAN latent space?," in *IEEE International Conference on Computer Vision*, 2019.

[125] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to edit the embedded images?," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[126] M. Huh, R. Zhang, J.-Y. Zhu, S. Paris, and A. Hertzmann, "Transforming and projecting images to class-conditional generative networks," in *European Conference on Computer Vision*, 2020.

[127] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *European Conference on Computer Vision*, 2020.

[128] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering interpretable GAN controls," in *Neural Information Processing Systems*, 2020.

[129] Y. Shen, C. Yang, X. Tang, and B. Zhou, "InterFaceGAN: Interpreting the disentangled face representation learned by GANs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[130] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[131] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, "StyleGAN-NADA: CLIP-guided domain adaptation of image generators," *ACM Transactions on Graphics*, 2021.

[132] R. Tzaban, R. Mokady, R. Gal, A. H. Bermano, and D. Cohen-Or, "Stitch it in time: GAN-based facial editing of real videos," *arXiv preprint arXiv:2201.08361*, 2022.

[133] Y. Alaluf, O. Patashnik, Z. Wu, A. Zamir, E. Shechtman, D. Lischinski, and D. Cohen-Or, "Third time's the charm? image and video editing with StyleGAN3," in *European Conference on Computer Vision*, 2023.

[134] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Neural Information Processing Systems*, 2021.

[135] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training Generative Adversarial Networks with limited data," in *Neural Information Processing Systems*, 2020.

[136] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[137] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least Squares Generative Adversarial Networks," in *IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

[138] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Neural Information Processing Systems*, 2017.

[139] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[140] W. Xia, Y. Zhang, Y. Yang, J.-H. Xue, B. Zhou, and M.-H. Yang, "GAN inversion: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3121–3138, 2023.

[141] J. Luo, Y. Xu, C. Tang, and J. Lv, "Learning inverse mapping by autoencoder based generative adversarial nets," in *International Conference on Neural Information Processing*, pp. 207–216, Springer, 2017.

[142] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or, "Face identity disentanglement via latent space mapping," *ACM Transactions on Graphics*, vol. 39, pp. 1 – 14, 2020.

[143] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "StyleGAN2 distillation for feed-forward image manipulation," in *European Conference on Computer Vision*, pp. 170–186, Springer, 2020.

[144] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a StyleGAN encoder for image-to-image translation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2021.

[145] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for StyleGAN image manipulation," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–14, 2021.

[146] L. Chai, J. Wulff, and P. Isola, "Using latent space regression to analyze and leverage compositionality in GANs," in *International Conference on Learning Representations*, 2021.

[147] A. Raj, Y. Li, and Y. Bresler, "GAN-based projector for faster recovery with convergence guarantees in linear inverse problems," in *IEEE International Conference on Computer Vision*, pp. 5602–5611, 2019.

[148] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code GAN prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3012–3021, 2020.

[149] G. Daras, A. Odena, H. Zhang, and A. G. Dimakis, "Your local GAN: Designing two dimensional local attention mechanisms for generative models," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14531–14539, 2020.

[150] D. Bau, H. Strobelt, W. Peebles, B. Zhou, J.-Y. Zhu, A. Torralba, *et al.*, "Semantic photo manipulation with a generative image prior," *ACM Transactions on Graphics*, vol. 38, no. 4, 2020.

[151] Z. Wu, D. Lischinski, and E. Shechtman, "Stylespace analysis: Disentangled controls for StyleGAN image generation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12863–12872, June 2021.

[152] O. K. Yüksel, E. Simsar, E. G. Er, and P. Yanardag, "LatentCLR: A contrastive learning approach for unsupervised discovery of interpretable directions," in *IEEE International Conference on Computer Vision*, pp. 14263–14272, 2021.

[153] B. Li, S. Cai, W. Liu, P. Zhang, M. Hua, Q. He, and Z. Yi, "Dystyle: Dynamic neural network for multi-attribute-conditioned style editing," in *IEEE Winter Conference on Applications of Computer Vision*, 2023.

[154] Y. Alaluf, O. Patashnik, and D. Cohen-Or, "Only a matter of style: Age transformation using a style-based regression model," *arXiv preprint arXiv:2102.02754*, 2021.

[155] Y. Wu, Y.-L. Yang, Q. Xiao, and X. Jin, "Coarse-to-fine: facial structure editing of portrait images via latent space classifications," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–13, 2021.

[156] M. Afifi, M. A. Brubaker, and M. S. Brown, "HistoGAN: Controlling colors of GAN-generated and real images via color histograms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[157] R. Saha, B. Duke, F. Shkurti, G. W. Taylor, and P. Aarabi, "Loho: Latent optimization of hairstyles via orthogonalization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1984–1993, 2021.

[158] S. Kwong, J. Huang, and J. Liao, "Unsupervised image-to-image translation via pre-trained StyleGAN2 network," *IEEE Transactions on Multimedia*, 2021.

[159] W. Jang, G. Ju, Y. Jung, J. Yang, X. Tong, and S. Lee, "StyleCariGAN: caricature generation via StyleGAN feature map modulation," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–16, 2021.

[160] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *IEEE International Conference on Computer Vision*, 2017.

[161] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Temporally coherent completion of dynamic video," *ACM Trans. Graphics*, vol. 35, no. 6, pp. 1–11, 2016.

[162] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon, "Unwrap mosaics: A new representation for video editing," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 1—11, 2008.

[163] Y. Kasten, D. Ofri, O. Wang, and T. Dekel, "Layered neural atlases for consistent video editing," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–12, 2021.

[164] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Trans. Graphics*, vol. 34, no. 6, pp. 1–9, 2015.

[165] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3D dense face alignment," in *European Conference on Computer Vision*, 2020.

[166] D. Rho, J. Cho, J. H. Ko, and E. Park, "Neural residual flow fields for efficient video representations," *arXiv preprint arXiv:2201.04329*, 2022.

[167] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

[168] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Neural Information Processing Systems*, 2020.

[169] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *European Conference on Computer Vision*, 2018.

[170] Y.-L. Liu, W.-S. Lai, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Learning to see through obstructions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[171] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systemss*, vol. 25, pp. 1097–1105, 2012.

[172] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.