WarpGAN: Warping-Guided 3D GAN Inversion with Style-Based Novel View Inpainting

Kaitao Huang[1], **Yan Yan**[1†], **Jing-Hao Xue**[2], **Hanzi Wang**[1]
[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, P.R. China
[2]Department of Statistical Science, University College London, UK
huangkt@stu.xmu.edu.cn, yanyan@xmu.edu.cn

Figure 1: **Visual examples**. Given a single input image (the first row), our WarpGAN synthesizes images from five novel views: front, right, left, top, and down (the second to the sixth rows).

Abstract

3D GAN inversion projects a single image into the latent space of a pre-trained 3D GAN to achieve single-shot novel view synthesis, which requires visible regions with high fidelity and occluded regions with realism and multi-view consistency. However, existing methods focus on the reconstruction of visible regions, while the generation of occluded regions relies only on the generative prior of 3D GAN. As a result, the generated occluded regions often exhibit poor quality due to the information loss caused by the low bit-rate latent code. To address this, we introduce the *warping-and-inpainting* strategy to incorporate image inpainting into 3D GAN inversion and propose a novel 3D GAN inversion method, **WarpGAN**. Specifically, we first employ a 3D GAN inversion encoder to project the single-view image into a latent code that serves as the input to 3D GAN. Then, we perform warping to a novel view using the depth map generated by 3D GAN. Finally, we develop a novel SVINet, which leverages the symmetry prior and multi-view image correspondence *w.r.t.* the same latent code to perform inpainting of occluded regions in the warped image. Quantitative and qualitative experiments demonstrate that our method consistently outperforms several state-of-the-art methods.

† Corresponding Author
Please visit the Project Page for visualizations and code.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

GANs [13] have made remarkable progress in synthesizing unconditional images. In particular, StyleGAN [20, 21] has achieved photorealistic quality on high-resolution images. Several extensions [15, 31, 36] leverage the latent space (i.e., the $\mathcal{W}$ space) to control semantic attributes (e.g., expression and age). However, these 2D GANs suffer from inferior control over geometrical aspects of generated images, leading to multi-view inconsistency for viewpoint manipulation.

Recently, with the development of neural radiance fields (NeRF) [27] in novel view synthesis (NVS), a variety of 3D GANs [2, 5, 6, 14, 29, 39, 41] have been proposed to integrate NeRF into style-based generation, resulting in remarkable success in generating highly realistic images. Based on it, 3D GAN inversion methods project a single image into the latent space of a pre-trained 3D GAN generator, obtaining a latent code. Hence, the viewpoint of the input image can be changed by altering the camera pose, and the image attributes can be easily edited by modifying the latent code. Unlike 2D GAN inversion, 3D GAN inversion aims to generate images that maintain both the faithfulness of the input view and the high quality of the novel views.

On the one hand, existing 3D GAN inversion methods rely only on the generative prior of 3D GANs for generating the occluded regions (i.e., the invisible regions in the input image) in the novel viewpoint, resulting in unfaithful reconstruction of occluded regions in complex scenarios. On the other hand, for 3D scene generation, several recent methods adopt a *warping-and-inpainting* strategy. They [11, 30, 35] first predict a depth map of a given image, and then warp the input image to novel camera viewpoints with the depth-based correspondence, followed by a 2D inpainting network to synthesize high-fidelity occluded regions of the warped images.

To address the inferior reconstruction capability of occluded regions in existing 3D GAN inversion methods, motivated by the success of the *warping-and-inpainting* strategy in 3D scene generation, we introduce image inpainting into 3D GAN inversion. Unfortunately, 3D GAN inversion is dedicated to training with single-view datasets, while the above 3D scene generation methods usually require multi-view datasets for training. This leads to two issues: (1) **multi-view inconsistency** due to the lack of 3D information (i.e., the real novel view image) to guide the inpainting process; (2) **the unavailability of ground-truth images** from novel views to compute the loss during model training.

In this paper, we propose a novel 3D GAN inversion method, **WarpGAN**, by integrating the *warping-and-inpainting* strategy into 3D GAN inversion. Specifically, we first train a 3D GAN inversion encoder, which projects the input image into a latent code w^+ (located in the latent space $\mathcal{W}^+$ of the 3D GAN generator). By feeding w^+ into 3D GAN, we compute the depth map of the input image for geometric warping and perform an initial filling of the occluded regions in the warped image. Subsequently, leveraging the symmetry prior [43, 45] and multi-view image correspondence *w.r.t.* the same latent code in 3D GANs, we train a style-based novel view inpainting network (SVINet). It can inpaint the occluded regions in the warped image from the original view to the novel view. Hence, we can synthesize plausible novel view images with multi-view consistency. To address the unavailability of ground-truth images, we re-warp the image in the novel view back to the original view and feed it to SVINet. Hence, the loss can be calculated between the inpainting result and the input image. Some visual examples obtained by WarpGAN are given in Fig. 1.

In summary, the contributions of this paper are as follows:

- We propose a novel 3D GAN inversion method, WarpGAN, which successfully introduces the *warping-and-inpainting* strategy into 3D GAN inversion, substantially enhancing the quality of occluded regions in novel view synthesis.
- We introduce a style-based novel view inpainting network, SVINet, by fully leveraging the symmetry prior and the same latent code generated by 3D GAN inversion, achieving multi-view consistency inpainting on the occluded regions of warped images in novel views.
- We perform extensive experiments to validate the superiority of WarpGAN, showing the great potential of the *warping-and-inpainting* strategy in 3D GAN inversion.

2 Related work

3D-Aware GANs. Recent advancements in 3D-Aware GANs [2, 5, 6, 14, 29, 39, 41] effectively combine the high-quality 2D image synthesis of StyleGAN [20, 21] with the multi-view synthesis

capability of NeRF [27], advancing high-quality image synthesis from 2D to 3D and enabling multi-view image generation. These methods typically employ a two-stage generation pipeline, where a low-resolution raw image and feature maps are rendered, followed by upsampling to high-resolution using 2D CNN layers. Such a way ensures geometric consistency across multiple views and achieves impressive photorealism. In this paper, we leverage EG3D [5] as our 3D-aware GAN architecture, which introduces a hybrid explicit-implicit 3D representation (known as the tri-plane).

GAN Inversion. Although recent 2D GAN inversion methods [42] have achieved promising editing performance, they suffer from severe flickering and inevitable multi-view inconsistency when editing 3D attributes (e.g., head pose) since the pretrained generator is not 3D-aware. Hence, 3D GAN inversion is developed to maintain multi-view consistency when rendering novel viewpoints. However, directly transferring 2D methods to 3D without effectively incorporating 3D information will inevitably lead to geometry collapse and artifacts.

Similar to 2D GAN inversion, 3D GAN inversion can be categorized into optimization-based methods and encoder-based methods. Some optimization-based methods [23, 43, 45] generate multiple pseudo-images from different viewpoints to facilitate optimization. For instance, HFGI3D [43] leverages visibility analysis to achieve pseudo-multi-view optimization; SPI [45] utilizes the facial symmetry prior to synthesize pseudo multi-view images; and Pose Opt. [23] simultaneously optimizes camera pose and latent codes. In addition, In-N-Out [44] optimizes a triplane for out-of-distribution object reconstruction and employs composite volume rendering. Encoder-based methods project the input image into the latent space of the 3D GAN generator and then employ the generative capacity of the 3D GAN to synthesize novel-view images, while fully utilizing the input image to reconstruct the visible regions of the novel-view images. For example, GOAE [46] computes the residual between the input image and the reconstructed image to complement the $\mathcal{F}$ space of the generator, and introduces an occlusion-aware mix tri-plane for novel-view image generation; Triplanenet [3] calculates an offset for the triplane based on the residual and proposes a facial symmetry prior loss; and Dual Encoder [4] employs two encoders (one for visible regions and the other for occluded regions) for inversion and introduces an occlusion-aware triplane discriminator to enhance both fidelity and realism.

Our method is intrinsically different from existing methods that rely heavily on 3D GAN generative priors to generate occluded regions. Our method introduces a novel inpainting network to fill the occluded regions, facilitating the generation of rich details.

Depth-based Warping for Single-shot Novel View Synthesis. Some 3D GAN inversion methods [23, 43, 45] use depth-based warping to synthesize pseudo multi-view images for optimization. SPI [45] warps the input image to an adjacent view for pseudo-supervision. Pose Opt. [23] warps the image from the canonical viewpoint to the input viewpoint to assist training. HFGI3D [43] utilizes a 3D GAN to fill the occluded regions of the warped image from the input view to novel views, synthesizing several pseudo novel-view images. However, these methods only rely on a 3D GAN to generate occluded regions, failing to achieve satisfactory results in occluded regions under complex scenarios.

Recently, some methods follow the *warping-and-inpainting* strategy on single-shot NVS for general scenes [11, 30, 35]. They first predict a depth map for the input image, then warp the input image to a novel view using the depth map, and finally perform inpainting on the occluded regions in the novel view. This way can effectively preserve the information of the input image while leveraging the powerful inpainting capability of 2D inpainting networks to generate reasonable content for occluded regions. Inspired by this strategy, we introduce a 2D inpainting network into 3D GAN inversion by effectively exploiting the symmetry prior and the latent code of the input image.

3 Methodology

3.1 Overview

As shown in Fig. 2, our WarpGAN consists of a 3D GAN inversion network (including a 3D GAN inversion encoder and a 3D-aware GAN) and a style-based novel view inpainting network (SVINet). First, we utilize a 3D GAN inversion encoder E_{w^+} to project the input image $\mathbf{I}$ into the latent space $\mathcal{W}^+$ of the 3D GAN generator, obtaining the latent code w^+. Based on this, we utilize a rendering decoder to render the depth map $\mathbf{D}$ of $\mathbf{I}$ and the novel view image $\hat{\mathbf{I}}^{w^+}_{novel}$. Under the guidance of the depth map $\mathbf{D}$, we warp the input image $\mathbf{I}$ from the original view c to the novel view c_{novel}, thereby obtaining the warped image $\mathbf{I}^{warp}_{c\to c_{novel}}$ and the occluded regions $\mathbf{M}^{o}_{c\to c_{novel}}$ of the input image in the

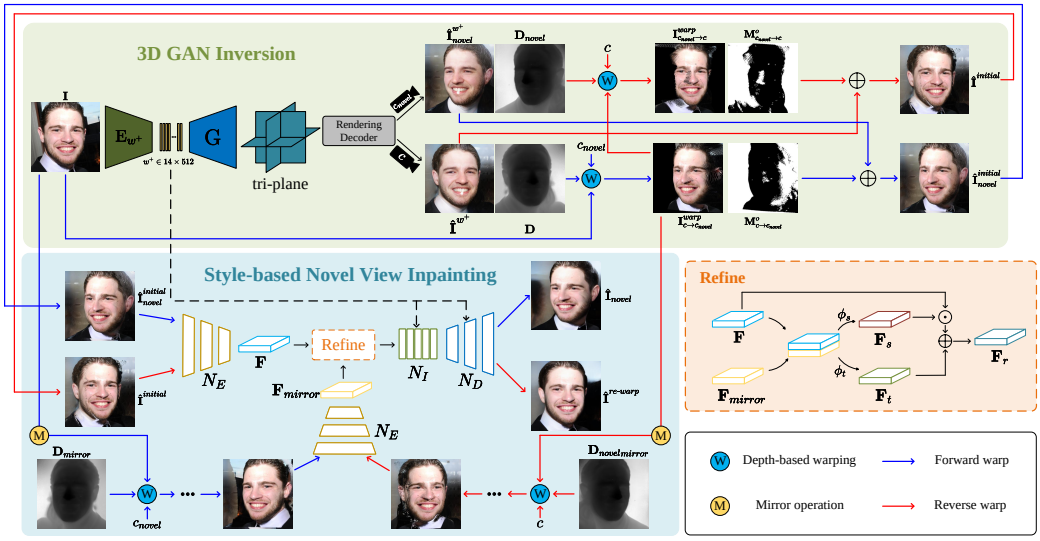

Figure 2: **Overview of our WarpGAN**, which consists of a 3D GAN inversion network and a style-based novel view inpainting network (SVINet). The "Forward warp" flow (blue arrows) illustrates the inference process of novel view synthesis. During model training, we also require the "Reverse warp" flow (red arrows) to warp the novel view image back to the original view for loss computation.

target view, that is,

$$\mathbf{I}^{warp}_{c\to c_{novel}}, \mathbf{M}^{o}_{c\to c_{novel}} = \text{warp}(\mathbf{I}; \mathbf{D}, \pi_{c\to c_{novel}}, K), \tag{1}$$

where $\pi_{c\to c_{novel}}$ is a relative camera pose between c and c_{novel}, K is the camera intrinsic matrix, and warp$(\cdot)$ is a geometric warping function [28, 35] which unprojects pixels of the input image $\mathbf{I}$ with its depth map $\mathbf{D}$ to the 3D space, and reprojects them based on $\pi_{c\to c_{novel}}$ and K.

Then, we use $\hat{\mathbf{I}}^{w^+}_{novel}$ to fill in the occluded regions of $\mathbf{I}^{warp}_{c\to c_{novel}}$, serving as the initial result $\hat{\mathbf{I}}^{initial}_{novel}$ for the occluded regions, which can be formulated as

$$\hat{\mathbf{I}}^{initial}_{novel} = \mathbf{I}^{warp}_{c\to c_{novel}} + \mathbf{M}^{o}_{c\to c_{novel}} \cdot \hat{\mathbf{I}}^{w^+}_{novel}. \tag{2}$$

Subsequently, the initial result $\hat{\mathbf{I}}^{initial}_{novel}$ is fed into SVINet for further inpainting, giving the final output $\hat{\mathbf{I}}_{novel}$ of WarpGAN. Notably, we employ symmetry-aware feature extraction and modulate the convolutions of the inpainting network with w^+ during the inpainting process. We also construct a style-based loss to ensure consistency between the generated image in the novel view and the original view image.

3.2 3D GAN Inversion Encoder

Similar to existing encoder-based 3D GAN inversion methods, our 3D GAN inversion encoder E_{w^+} projects an input image $\mathbf{I}$ with the camera pose c into the latent space $\mathcal{W}^+$ of the pre-trained 3D GAN, obtaining the latent code $w^+ = \text{E}_{w^+}(\mathbf{I})$. Then, we leverage the generator $\text{G}(\cdot)$ of 3D GAN to generate the tri-plane and use the rendering decoder $\mathcal{R}$ to render images at specified camera poses. Based on above, we perform image reconstruction $\hat{\mathbf{I}}^{w^+} = \mathcal{R}(\text{G}(w^+), c)$ by specifying the camera pose as c. In this way, we obtain the novel view image $\hat{\mathbf{I}}^{w^+}_{novel}$ corresponding to the novel camera pose c_{novel}. Under the principles of NeRF, we replace the color of the sampling points with the distance to the camera during the rendering process, obtaining the depth maps $\mathbf{D}$ and $\mathbf{D}_{novel}$. More implementation details can be found in the **Appendix**.

Inspired by GOAE [46], we employ a pyramid-structured Swin-Transformer [26] as the backbone of the encoder, based on which we leverage feature layers at different scales to generate latent codes at various levels.

Since our dataset contains only single-view images, we train E_{w^+} using a reconstruction loss $\mathcal{L}_{w^+}$, which includes a pixel-wise (MSE) loss $\mathcal{L}_2$, a perceptual loss $\mathcal{L}_{\text{LPIPS}}$ [48], and an identity loss $\mathcal{L}_{\text{ID}}$ with a pre-trained ArcFace network [12]:

$$\mathcal{L}_{w^+}(\hat{\mathbf{I}}^{w^+}, \mathbf{I}) = \lambda_2 \mathcal{L}_2(\hat{\mathbf{I}}^{w^+}, \mathbf{I}) + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}(\hat{\mathbf{I}}^{w^+}, \mathbf{I}) + \lambda_{\text{ID}}^{w^+} \mathcal{L}_{\text{ID}}(\hat{\mathbf{I}}^{w^+}, \mathbf{I}), \tag{3}$$

where λ_2, λ_{LPIPS}, and $\lambda_{\text{ID}}^{w^+}$ denote the loss weights for $\mathcal{L}_2$, $\mathcal{L}_{\text{LPIPS}}$, and $\mathcal{L}_{\text{ID}}$, respectively.

3.3 Style-Based Novel View Inpainting Network (SVINet)

Due to the existence of occluded regions in the novel view, the warped image contains "holes" (see Fig. 2 for an illustration). To generate high-quality novel-view images, we propose a style-based novel view inpainting network (SVINet) to fill in the "holes" in the warped image.

As shown in Fig. 2, our SVINet follows the traditional "encode-inpaint-decoder" architecture [10, 24, 37], consisting of three sub-networks: N_E, N_I, and N_D. Technically, N_E is first used to extract features from the model input while performing downsampling. Then, the inpainting operation is performed in the feature space by using N_I. Finally, N_D is used to upsample the features to obtain the inpainted image.

3.3.1 Symmetry-Aware Feature Extraction

We first use the novel-view image $\hat{\mathbf{I}}_{novel}^{w^+}$ obtained from 3D GAN inversion to fill in the occluded regions in the warped image $\mathbf{I}_{c \to c_{novel}}^{warp}$ (Eq. (1)), resulting in an initial inpainting result $\hat{\mathbf{I}}_{novel}^{initial}$ (Eq. (2)). We then feed $\hat{\mathbf{I}}_{novel}^{initial}$ into N_E to obtain the feature $\mathbf{F}$. In addition, we also propose to leverage the facial symmetry [43, 45] by warping the mirrored input image $\mathbf{I}_{mirror}$ to the target view c_{novel}, obtaining $\mathbf{I}_{mirror}{}_{c_{mirror} \to c_{novel}}^{warp}$. The mirrored image is then processed in the same manner as described above and fed into N_E to obtain the mirror feature $\mathbf{F}_{mirror}$.

Subsequently, we utilize $\mathbf{F}$ and $\mathbf{F}_{mirror}$ to predict the scale map $\mathbf{F}_s$ and the translation map $\mathbf{F}_t$, which can be used to refine $\mathbf{F}$ via featurewise linear modulation (FiLM) [32], obtaining $\mathbf{F}_r$, that is,

$$\begin{aligned} \{\mathbf{F}_s, \mathbf{F}_t\} &= \{\phi_s([\mathbf{F}, \mathbf{F}_{mirror}]_1), \phi_t([\mathbf{F}, \mathbf{F}_{mirror}]_1)\}, \\ \mathbf{F}_r &= \mathbf{F}_s \odot \mathbf{F} + \mathbf{F}_t, \end{aligned} \tag{4}$$

where ϕ_s and ϕ_t are convolutional neural networks; $[,]_1$ denotes concatenation along the 1th dimension, i.e., the channel dimension; '$\odot$' denotes the Hadamard product.

Next, $\mathbf{F}^r$ is successively fed into N_I and N_D to obtain the inpainting result $\hat{\mathbf{I}}_{novel}$.

3.3.2 Style-Based Inpainting

Inpainting networks typically rely on the information of the input image to fill in the missing regions. However, due to the limited information contained in single-view images, using only this information for inpainting may lead to the issue of multi-view inconsistency. To address the consistency issue, motivated by the fact that images of the same object from different viewpoints share the same latent code in 3D GANs, we introduce the latent code to control the image inpainting process.

Technically, we modulate the convolutions [21, 24] in the "inpaint" and "decoder" parts of the inpainting network using the latent code w^+ obtained from E_{w^+}. This modulation of the convolutions facilitates us to control the inpainting process for occluded regions, achieving multi-view consistency in the generated images.

Specifically, we first employ a mapping function $\mathcal{A}$ to obtain the style code $s = \mathcal{A}(w^+)$. Then the weights of the convolutions w are modulated as

$$\begin{aligned} w'_{ijk} &= s_i \cdot w_{ijk}, \\ w''_{ijk} &= w'_{ijk} \Big/ \sqrt{\textstyle\sum_{i,k} {w'_{ijk}}^2 + \epsilon}, \end{aligned} \tag{5}$$

where w'' denotes the final modulated weights; s_i is the scale corresponding to the ith input feature map; j and k enumerate the output feature maps and spatial footprint of the convolution, respectively.

3.3.3 Training strategy

Real data. Since our real dataset contains only single-view images, no target-view images can be used to compute the loss and update the model parameters when synthesizing images from novel views. To address this, we propose to re-warp the warped image from the novel view back to the original view, and then compute the loss between the inpainting result and the input image.

Specifically, for the input image $\mathbf{I}$, we first warp it to the novel view c_{novel} to obtain $\mathbf{I}^{warp}_{c\to c_{novel}}$, and then inpaint it using SVINet to get $\hat{\mathbf{I}}_{novel}$. Next, we re-warp $\mathbf{I}^{warp}_{c\to c_{novel}}$ back to the source view c and inpaint it again to obtain $\hat{\mathbf{I}}^{re\text{-}warp}$. Based on the above, given the input image $\mathbf{I}$, we obtain two inpainted images $\hat{\mathbf{I}}_{novel}$ and $\hat{\mathbf{I}}^{re\text{-}warp}$ for loss computation.

Synthetic data. In addition to real data, we also utilize synthetic data to assist in training our model. We sample a latent code w_{synth} from the latent space of 3D GAN and generate two images $\mathbf{I}^{synth}_s$ and $\mathbf{I}^{synth}_t$ from different viewpoints. We then warp $\mathbf{I}^{synth}_s$ from the source view to the target view and input it into SVINet to obtain the inpainted image $\hat{\mathbf{I}}^{synth}_t$. Finally, we compute the loss between $\hat{\mathbf{I}}^{synth}_t$ and $\mathbf{I}^{synth}_t$.

Loss function. Our loss function consists of three components: the reconstruction loss, the consistency loss, and the adversarial loss. The reconstruction loss $\mathcal{L}_{\text{rec}}$ includes the pixel-wise MAE loss $\mathcal{L}_1$, the perceptual loss $\mathcal{L}_{\text{P}}$ [37], and the identity loss $\mathcal{L}_{\text{ID}}$ [12]:

$$\mathcal{L}_{\text{rec}}(\hat{\mathbf{I}},\mathbf{I}) = \lambda_1\mathcal{L}_1(\hat{\mathbf{I}}-\mathbf{I}) + \lambda_{\text{P}}\mathcal{L}_{\text{P}}(\hat{\mathbf{I}},\mathbf{I}) + \lambda_{\text{ID}}\mathcal{L}_{\text{ID}}(\hat{\mathbf{I}},\mathbf{I}), \tag{6}$$

where λ_1, λ_{P}, and λ_{ID} denote the loss weights for $\mathcal{L}_1$, $\mathcal{L}_{\text{P}}$, and $\mathcal{L}_{\text{ID}}$, respectively; $\hat{\mathbf{I}}$ and $\mathbf{I}$ represent the input image and the generated image, respectively.

To ensure multi-view consistency, we introduce the consistency loss $\mathcal{L}_{\text{c}}$, which computes the MSE between the latent codes of the original image and the inpainted image. This loss is used to control the multi-view consistency of the generated images:

$$\mathcal{L}_{\text{c}}(\hat{\mathbf{I}},\mathbf{I}) = ||\text{E}_{w^+}(\hat{\mathbf{I}}),\text{E}_{w^+}(\mathbf{I})||_2. \tag{7}$$

To further enhance the quality of the inpainted images, we also use an adversarial loss:

$$\mathcal{L}^G_{\text{adv}} = -\mathbb{E}[\log(D(\hat{x}))], \tag{8}$$

$$\mathcal{L}^D_{\text{adv}} = -\mathbb{E}[\log(D(x))] - \mathbb{E}[\log(1-D(\hat{x}))] + \gamma\mathbb{E}[||\nabla D(x)||_2], \tag{9}$$

where x denotes the real and synthetic images (i.e., $\mathbf{I}$ and $\mathbf{I}^{synth}_t$); $\hat{x}$ represents the inpainted images (i.e., $\hat{\mathbf{I}}_{novel}$, $\hat{\mathbf{I}}^{re\text{-}warp}$, and $\hat{\mathbf{I}}^{synth}_t$); D denotes the discriminator [10, 24, 37].

In summary, the loss function for SVINet can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{SVINet}} &= \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}([\hat{\mathbf{I}}^{re\text{-}warp},\hat{\mathbf{I}}^{synth}_t]_0,[\mathbf{I},\mathbf{I}^{synth}_t]_0)\\ &+ \lambda_{\text{c}}\mathcal{L}_{\text{c}}([\hat{\mathbf{I}}_{novel},\hat{\mathbf{I}}^{re\text{-}warp},\hat{\mathbf{I}}^{synth}_t]_0,[\mathbf{I},\mathbf{I},\mathbf{I}^{synth}_t]_0) + \lambda_{\text{adv}}\mathcal{L}^G_{\text{adv}},\end{aligned} \tag{10}$$

where $[,]_0$ denotes concatenation along the 0-th dimension (i.e., the batch dimension); λ_{rec}, λ_{c}, and λ_{adv} denote the loss weights for $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{c}}$, and $\mathcal{L}^G_{\text{adv}}$, respectively.

4 Experiments

4.1 Experimental Settings

Datasets. Our experiments mainly focus on face datasets. We use the FFHQ dataset [20] and 100K pairs of synthetic data for training. The synthetic pairs $\{\mathbf{I}^{synth}_s,\mathbf{I}^{synth}_t\}$ are generated from EG3D [5], sharing the same latent code w_{synth} but rendered with different camera poses. To evaluate the generalization ability of our method, we employ the CelebA-HQ dataset [19] and the multi-view MEAD dataset [40] for testing. We preprocess the images in the datasets and extract their camera poses in the same manner as [5].

Implementation Details. For all experiments, we employ the EG3D [5] generator pre-trained on FFHQ. For the 3D GAN inversion encoder E_{w^+}, we set the batch size to 4 and train it for 500K

Table 1: **Comparisons with state-of-the-art methods** on the CelebA-HQ and MEAD datasets.

Category	Method	CelebA-HQ		MEAD						Time (s)↓
		FID↓	ID↑	LPIPS ↓		FID ↓		ID ↑		
				±30°	±60°	±30°	±60°	±30°	±60°	
Optimization	SG2 $\mathcal{W}^+$	26.09	0.7369	0.2910	0.3372	39.30	64.47	0.7992	0.7533	43.72
	PTI	25.70	0.7616	0.2771	0.3341	44.23	66.00	0.8089	0.7582	62.65
	Pose Opt.	29.04	0.7500	0.2990	0.3428	52.25	73.23	0.7954	0.7405	91.60
	HFGI3D	24.30	0.7641	0.2775	0.3494	51.24	79.81	0.8019	0.7370	264.5
Encoder	pSp	38.46	0.7375	0.3116	0.3720	65.21	94.34	0.7900	0.7401	0.05430
	GOAE	35.41	0.7498	0.2818	0.3453	59.69	86.23	0.8109	0.7370	0.07999
	Triplanenet	32.65	0.7706	0.3379	0.4103	76.62	130.55	0.8059	0.7135	0.1214
	Ours	**19.12**	**0.7882**	**0.2490**	**0.3008**	**38.15**	**64.01**	**0.8315**	**0.7741**	0.08390

Figure 3: **Comparisons of novel view synthesis** on the CelebA-HQ dataset between our WarpGAN and several state-of-the-art methods.

iterations on the FFHQ dataset. We use the Ranger optimizer, which combines Rectified Adam [25] with the Lookahead technique [47], with learning rates of 1e-4 for E_{w^+}. The values of λ_2, λ_{LPIPS}, and $\lambda_{\text{ID}}^{w^+}$ in Eq. (3) are set to 1.0, 0.8, and 0.1. For SVINet, we set the batch size to 2 and train it for 300K iterations on both the FFHQ dataset and synthetic data pairs. For the novel view camera poses during the training process, we sample from the camera poses of the pose-rebalanced FFHQ dataset [5]. We use the Adam optimizer [22], with learning rates of 1e-3 and 1e-4 for the SVINet and discriminator, respectively. The values of λ_1, λ_{P}, and λ_{ID} in Eq. (6) are set to 10.0, 30.0, and 0.1, respectively. The values of λ_{rec}, λ_{c}, and λ_{adv} in Eq. (10) are set to 1.0, 0.1, and 10.0, respectively.

Baselines. We compare our WarpGAN with several 3D GAN inversion methods, including optimization-based methods (such as SG2 $\mathcal{W}^+$ [1], PTI [34], Pose Opt. [23], and HFGI3D [43]) and encoder-based methods (such as pSp [33], GOAE [46], Triplanenet [3], and Dual Encoder [4]). Note that Dual Encoder employs a 3D GAN other than EG3D and removes the background during training. This is different from our experimental setup, we only compare it in the qualitative analysis.

Evaluation metrics. We perform novel view synthesis evaluation on the CelebA-HQ dataset and the MEAD dataset. For the CelebA-HQ dataset, we compute the Fréchet Inception Distance (FID) [17] and ID similarity [12] between the original images and the novel view images. For the multi-view MEAD dataset, each person includes five face images with increasing yaw angles (front, $\pm 30°$, and $\pm 60°$). We use the front image as input and synthesize the other four views. We then compute the LPIPS [48], FID, and ID similarity between the synthesized images and their corresponding ground-truth images. The inference times (Time) in Table 1 are measured on a single Nvidia GeForce RTX 4090 GPU.

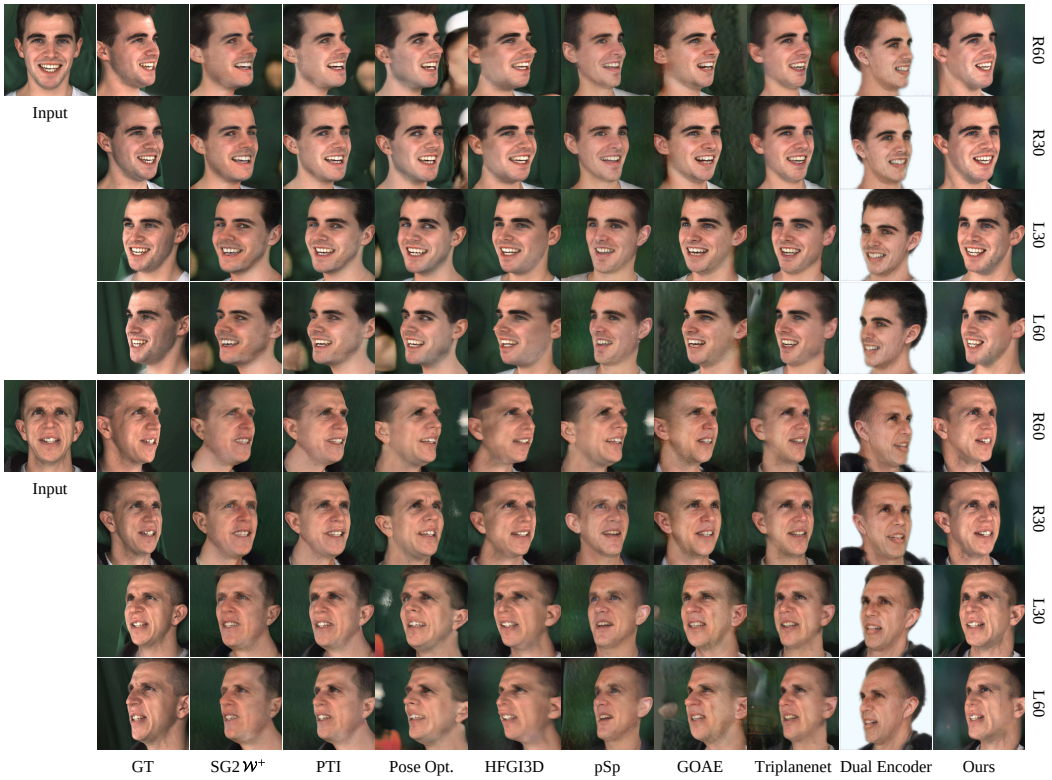

Figure 4: **Comparisons of different methods** on the MEAD dataset for synthesizing images of the other four views (R60, R30, L30, and L60) using the front image as input.

4.2 Comparisons with State-of-the-Art Methods

Quantitative Evaluation. As shown in Table 1, we provide the performance of different methods on the CelebA-HQ dataset and the MEAD dataset. It can be clearly observed that optimization-based methods achieve better performance than encoder-based methods, but at the cost of significantly higher inference times. Among them, HFGI3D, which performs optimization twice using PTI (once for filling the occluded regions of warped images and once for multi-view optimization), shows substantial performance improvement but suffers from slow inference times. In contrast, our WarpGAN, which has an inference time comparable to encoder-based methods, surpasses the performance of optimization-based methods. The excellent performance on the MEAD dataset demonstrates that our method is capable of effectively preserving multi-view consistency.

Qualitative Evaluation. We provide visualization results of novel view synthesis in Fig. 3 and Fig. 4. By successfully integrating the *warping-and-inpainting* strategy into 3D GAN inversion, our method can better preserve facial details and generate more reasonable occluded regions. Moreover, our method is capable of maintaining 3D consistency in novel views more naturally.

4.3 Ablation Studies

Table 2: **Ablation** on different components of our WarpGAN.

Name	Model	FID ↓	ID ↑
A	E_{w^+}	36.07	0.7437
B	w/o SVINet	29.28	0.7735
C	w/o Mod_{w^+} & $\mathcal{L}_c$	19.71	0.7879
D	w/o Mod_{w^+}	19.47	0.7880
E	w/o symmetry	20.04	0.7825
F	w/o synth data	19.18	0.7880
G	Full Model	**19.12**	**0.7882**

To investigate the contributions of key components in our method, we conduct ablation studies. In Table 2, we compare the quality of novel view synthesis using different model variants on the CelebA-HQ dataset.

Comparing "B" and "G" clearly demonstrates the significant role of SVINet in inpainting occluded regions. Comparing "C", "D", and "G" shows that modulating the convolutions of SVINet with w^+ and incorporating $\mathcal{L}_c$ enhance the performance of our method. Comparing "E"

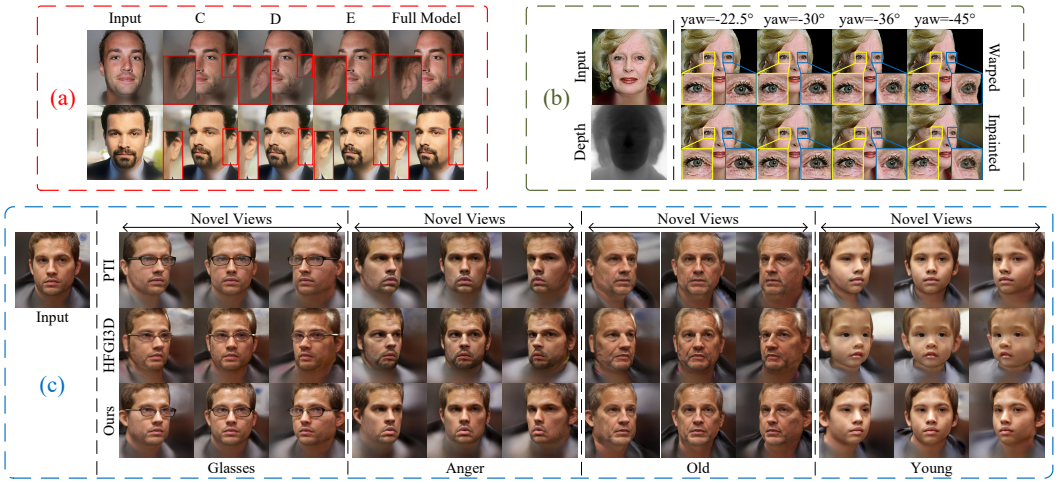

Figure 5: (a) **Qualitative comparisons** of the Full Model with model variants "C", "D", and "E"; (b) **Some failure cases**; (c) **Comparisons of image attribute editing effects** with PTI and HFGI3D.

and "G" indicates that leveraging facial symmetry prior helps generate occluded regions in novel views. Comparing "F" and "G" reveals that training with synthetic data slightly improves the quality of novel view synthesis. We also qualitatively compare "C", "D", "E", and "G" (Full Model) in Fig. 5(a). Incorporating the latent code to control the inpainting process of SVINet and the symmetry prior can provide more information, reduce blurring and artifacts, and generate more detailed results.

4.4 Editing Application

Since our WarpGAN achieves novel view synthesis by inpainting warped images, the visible parts of the novel view images are minimally affected by the latent code. Consequently, manipulating the latent code alone does not enable attribute editing of the image. To address this issue, similar to HFGI3D [43], we utilize WarpGAN to synthesize a series of novel view images, which are then fed into PTI [34] for optimization. This process yields an optimized latent code w^{+}_{opt} and a fine-tuned 3D GAN generator. In this way, attribute editing of the input image and novel view rendering can be achieved by editing w^{+}_{opt} [15, 31, 36] and modifying the camera pose c. As shown in Fig. 5(c), we perform attribute editing on the input image for four attributes: "Glasses", "Anger", "Old", and "Young", and compare the results with those from PTI and HFGI3D. It can be observed that the edited images obtained by using multi-view images synthesized by WarpGAN for optimization assistance exhibit higher fidelity and appear more natural.

5 Conclusion

In this paper, motivated by the achievement of the *warping-and-inpainting* strategy in 3D scene generation, we successfully integrate image inpainting with 3D GAN inversion and propose a novel 3D GAN inversion method, WarpGAN, for high-quality novel view synthesis from a single image. Our WarpGAN consists of a 3D GAN inversion network and SVINet. Specifically, we first obtain the depth of the input image using 3D GAN inversion, then apply depth-based warping to the input image to obtain the warped image, and finally use SVINet to fill in the occluded regions of the warped image. Notably, our SVINet leverages symmetry prior and the latent code for multi-view consistency inpainting. Extensive qualitative and quantitative experiments demonstrate that our method outperforms existing state-of-the-art optimization-based and encoder-based methods.

Limitations. Due to the inevitable errors in the depth map [11, 30, 35], the warped image sometimes become unreliable, which in turn prevents our SVINet from eliminating such artifacts. As illustrated in Fig. 5(b), when the angle variation is small, SVINet can alleviate the deformation of the eyes. However, as the angle of change increases, the output of SVINet deteriorates.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62372388 and Grant U21A20514, the Major Science and Technology Plan Project on the Future Industry Fields of Xiamen City under Grant 3502Z20241029 and Grant 3502Z20241027, and the Fundamental Research Funds for the Central Universities under Grant 20720240076 and Grant ZYGX2021J004.

References

[1] R. Abdal, Y. Qin, and P. Wonka. Image2StyleGAN++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020.

[2] S. An, H. Xu, Y. Shi, G. Song, U. Y. Ogras, and L. Luo. Panohead: Geometry-aware 3d full-head synthesis in 360deg. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20950–20959, 2023.

[3] A. R. Bhattarai, M. Nießner, and A. Sevastopolsky. Triplanenet: An encoder for eg3d inversion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3055–3065, 2024.

[4] B. B. Bilecen, A. Gökmen, and A. Dundar. Dual encoder GAN inversion for high-fidelity 3d head reconstruction from single images. *Advances in Neural Information Processing Systems*, pages 87357–87385, 2024.

[5] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.

[6] E. R. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021.

[7] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] L. Chi, B. Jiang, and Y. Mu. Fast fourier convolution. *Advances in Neural Information Processing Systems*, pages 4479–4488, 2020.

[9] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[10] T. Chu, J. Chen, J. Sun, S. Lian, Z. Wang, Z. Zuo, L. Zhao, W. Xing, and D. Lu. Rethinking fast fourier convolution in image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23195–23205, 2023.

[11] J. Chung, S. Lee, H. Nam, J. Lee, and K. M. Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

[12] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

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

[14] J. Gu, L. Liu, P. Wang, and C. Theobalt. StyleNeRF: A style-based 3d-aware generator for high-resolution image synthesis. In *Proceedings of International Conference on Learning Representations*, 2022.

[15] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. GANSpace: Discovering interpretable GAN controls. *Advances in Neural Information Processing Systems*, pages 9841–9850, 2020.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] 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. *Advances in Neural Information Processing Systems*, 30, 2017.

[18] J. T. Kajiya and B. P. Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, pages 165–174, 1984.

[19] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[20] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[21] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

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

[23] J. Ko, K. Cho, D. Choi, K. Ryoo, and S. Kim. 3d GAN inversion with pose optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2967–2976, 2023.

[24] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10758–10768, 2022.

[25] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of International Conference on Learning Representations*, 2020.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[27] 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. *Communications of the ACM*, pages 99–106, 2021.

[28] S. Niklaus and F. Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020.

[29] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman. StyleSDF: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022.

[30] H. Ouyang, K. Heal, S. Lombardi, and T. Sun. Text2immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242*, 2023.

[31] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.

[32] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021.

[34] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics*, pages 1–13, 2022.

[35] J. Seo, K. Fukuda, T. Shibuya, T. Narihira, N. Murata, S. Hu, C.-H. Lai, S. Kim, and Y. Mitsufuji. Genwarp: Single image to novel views with semantic-preserving generative warping. *Advances in Neural Information Processing Systems*, 2024.

[36] 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*, pages 2004–2018, 2020.

[37] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022.

[38] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics*, pages 1–14, 2021.

[39] A. Trevithick, M. Chan, T. Takikawa, U. Iqbal, S. De Mello, M. Chandraker, R. Ramamoorthi, and K. Nagano. What you see is what you GAN: Rendering every pixel for high-fidelity geometry in 3d GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22765–22775, 2024.

[40] K. Wang, Q. Wu, L. Song, Z. Yang, W. Wu, C. Qian, R. He, Y. Qiao, and C. C. Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *Proceedings of European Conference on Computer Vision*, pages 700–717, 2020.

[41] Y. Wu, J. Zhang, H. Fu, and X. Jin. Lpff: A portrait dataset for face generators across large poses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20327–20337, 2023.

[42] 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*, pages 3121–3138, 2022.

[43] J. Xie, H. Ouyang, J. Piao, C. Lei, and Q. Chen. High-fidelity 3d GAN inversion by pseudo-multi-view optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2023.

[44] Y. Xu, Z. Shu, C. Smith, S. W. Oh, and J.-B. Huang. In-n-out: Faithful 3d GAN inversion with volumetric decomposition for face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7225–7235, 2024.

[45] F. Yin, Y. Zhang, X. Wang, T. Wang, X. Li, Y. Gong, Y. Fan, X. Cun, Y. Shan, C. Oztireli, et al. 3d GAN inversion with facial symmetry prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 342–351, 2023.

[46] Z. Yuan, Y. Zhu, Y. Li, H. Liu, and C. Yuan. Make encoder great again in 3d GAN inversion through geometry and occlusion-aware encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2437–2447, 2023.

[47] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton. Lookahead optimizer: k steps forward, 1 step back. *Advances in Neural Information Processing Systems*, 32, 2019.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the proposed WarpGAN method, its components, and the improvements in novel view synthesis, accurately reflecting the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our method in Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not involve theoretical results or their proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the network architecture and the training strategy in Sec. 3, the dataset usage and hyperparameter settings in Sec. 4.1, and additional details in the Appendix, which fully disclose the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submit the code in the supplementary material, and all the datasets used are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all the training and test details, including datasets, implementation details, baselines, and evaluation metrics in Sec. 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our task is image generation rather than prediction. Same as existing 3D GAN inversion methods, we use metrics such as FID, ID, and LPIPS, which do not include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the execution times for various methods in Table 1 and specify the computational resources used in Sec. 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer:[Yes]

    Justification: We discuss the societal impacts in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We describe the potential risks in the Appendix.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite all the papers involved in our work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not introduce new assets in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper neither involves crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Additional Architecture Details

Detailed Structure of SVINet. LaMa [37] introduces fast Fourier convolutions (FFCs) [8] into image inpainting, achieving a receptive field that covers the whole image even in the early network layers. Such a way can facilitate the inpainting of large missing areas. To effectively fill in the occluded regions of the warped image, our SVINet is built upon the framework of LaMa and consists of three sub-networks: N_E, N_I, and N_D. For an input image with the size of 512×512, N_E includes 3 downsampling convolutional layers that downsample the input image to a feature map with the size of 64×64; N_I contains 9 FFC residual blocks, each of which consists of two FFCs and a residual connection, for inpainting; and N_D consists of 3 upsampling convolutional layers to upsample the image resolution back to the size of 512×512. The convolutions in N_I and N_D are modulated by the latent code w^+ from the 3D GAN inversion encoder E_{w^+}. Note that, each FFC contains three convolutional branches and one spectral transform branch, and the convolutions within the spectral transform are also modulated, as shown in Fig. 6.

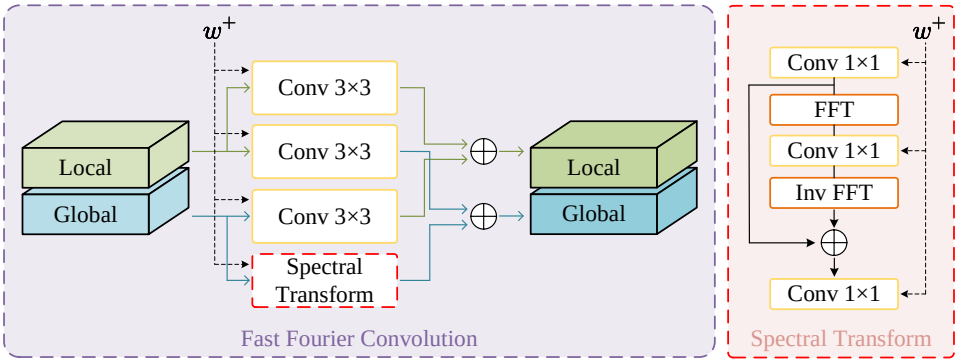

Figure 6: **The detailed structure** of fast Fourier convolution modulated by the latent code w^+.

B Additional Implementation Details

B.1 Principles of Neural Radiance Fields

Neural Radiance Fields (NeRF) [27] employs a fully-connected deep network, which maps a 3D spatial location $\mathbf{x}$ and a viewing direction $\mathbf{d}$ to color $\mathbf{c}$ and density σ, to represent a scene. By querying $\mathbf{x}$ and $\mathbf{d}$ along camera rays and applying classical volume rendering techniques [18], the color and density information can be projected into a 2D image. Specifically, for each projected ray $\mathbf{r}$ corresponding to a given pixel, N_s points (denoted as $\{t_i\}_{i=1}^{N_s}$) are sampled along the ray. For each sampled point, the estimated color and density are represented as $\mathbf{c}_i$ and σ_i, respectively. The RGB value $C(\mathbf{r})$ for each ray can then be computed via volumetric rendering as follows:

$$C(\mathbf{r}) = \sum_{i=1}^{N_s} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \tag{11}$$

where $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$, and $\delta_i = t_{i+1} - t_i$ denotes the distance between adjacent samples.

Similarly, if we replace the color c_i of each sampled point with the distance t_i from the sampling point to the camera during volumetric rendering, the depth $d(\mathbf{r})$ along each ray can be obtained as

$$d(\mathbf{r}) = \sum_{i=1}^{N_s} T_i(1 - \exp(-\sigma_i \delta_i))t_i. \tag{12}$$

B.2 Multi-View Optimization for Editing

Our WarpGAN synthesizes novel view images not only based on the results of 3D GAN inversion but also relies on the warping results of the input image. Thus, only modifying the latent code within our method is difficult to achieve desirable editing effects. Inspired by HFGI3D [43], we employs WarpGAN to generate N novel view images $\{\mathbf{I}_i\}_{i=1}^N$ corresponding to N different camera poses $\{c_i\}_{i=1}^N$ to assist the optimization process of PTI [34], denoted as **WarpGAN-Opt**.

Specifically, for a single input image $\mathbf{I}$ with the camera pose c, we first employ an optimization-based GAN inversion method [1] to jointly optimize the latent code w^+ and the noise vector n in the 3D GAN generator:

$$w^+_{opt}, n = \arg\min_{w^+,n} \ \mathcal{L}_2(\mathcal{R}(\mathrm{G}(w^+, n; \theta), c), \mathbf{I}) + \lambda_n \mathcal{L}_n(n), \tag{13}$$

where $\mathcal{L}_n$ is a noise regularization term and λ_n is a hyperparameter [34].

Subsequently, we fix the optimized latent code w^+_{opt} and fine-tune the 3D GAN generator based on the input image $\mathbf{I}$ and a series of novel view images synthesized by our WarpGAN:

$$\theta^* = \arg\min_{\theta} \ \mathcal{L}_{\mathrm{G}}(\mathcal{R}(\mathrm{G}(w^+_{opt}; \theta), c), \mathbf{I}) + \lambda_{mv} \sum_{i}^{N} \mathcal{L}_{\mathrm{G}}(\mathcal{R}(\mathrm{G}(w^+_{opt}; \theta), c_i), \mathbf{I}_i), \tag{14}$$

$$\mathcal{L}_{\mathrm{G}} = \lambda^{\mathrm{G}}_2 \mathcal{L}_2 + \lambda^{\mathrm{G}}_{\mathrm{LPIPS}} \mathcal{L}_{\mathrm{LPIPS}}, \tag{15}$$

where λ_{mv} is set to 1.0; both λ^{G}_2 and $\lambda^{\mathrm{G}}_{\mathrm{LPIPS}}$ are set to 1.0.

After the aforementioned process, we obtain the optimized latent code w^+_{opt} and the 3D GAN generator with tuned weights θ^*. To generate attribute-edited images from different viewpoints, we simply modify w^+_{opt} [31, 36], specify the desired camera pose c_{novel}, and feed them into the 3D GAN to obtain the edited image $\hat{\mathbf{I}}^{edit}_{novel}$ in the novel view, that is,

$$\hat{\mathbf{I}}^{edit}_{novel} = \mathcal{R}(\mathrm{G}(w^+_{opt} + \alpha \mathbf{n}_{att}; \theta^*), c_{novel}), \tag{16}$$

where $\mathbf{n}_{att}$ denotes a specific direction for attribute editing and α is a scaling factor.

C Broader Impacts

Our proposed method, which enables novel view synthesis and attribute editing of faces from a single image, holds the potential to significantly impact various fields such as film, gaming, augmented reality (AR), and virtual reality (VR). However, it also raises concerns regarding privacy and ethics, particularly the risk of generating "deep fakes". We emphasize the necessity of implementing robust safeguards to ensure the responsible and ethical application of this technology, thereby minimizing the risk of misuse.

D Additional Qualitative Results

Additional Qualitative Evaluation. We provide more visual comparisons between our WarpGAN and several state-of-the-art methods in Fig. 7. In addition, since we utilize multi-view images synthesized by WarpGAN to assist 3D GAN inversion optimization for editing, we also include comparisons with this optimization-based method (WarpGAN-Opt). We can see that, due to the limitations of the low bit-rate latent code, WarpGAN-Opt loses some detail compared with WarpGAN. However, by leveraging the high-quality novel view images synthesized by WarpGAN, WarpGAN-Opt achieves higher fidelity and realism in novel view synthesis than other optimization-based methods. From the figure, it can be observed that our method outperforms Dual Encoder [4]. However, since our method relies on the visible regions of the input image in the novel view to inpaint occluded regions, our method degrades to a typical encoder-based 3D GAN inversion when the view change is large and the visible region is small. In contrast, Dual Encoder focuses on high-fidelity 3D head reconstruction and thus offers greater flexibility in terms of view changes.

Additional Attribute Editing Results. To more comprehensively demonstrate the capability of our method in image attribute editing, we provide additional attribute editing results in Fig. 8. Specifically,

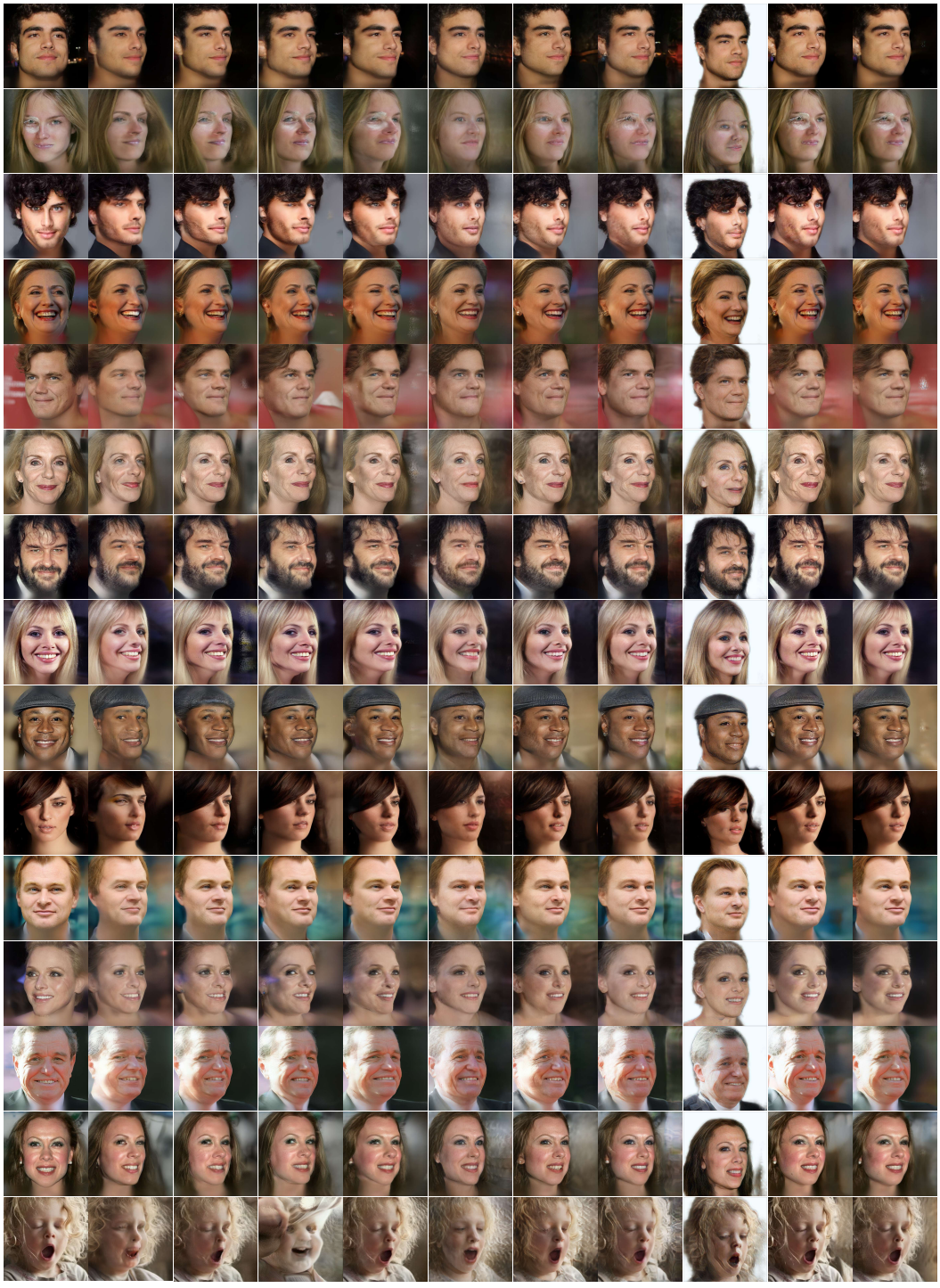

Figure 7: **Qualitative comparisons** between our WarpGAN and several state-of-the-art methods.

we employ InterFaceGAN [36] for editing the "Anger", "Old", and "Young" attributes, and utilize the text-guided semantic editing method StyleCLIP [31] for editing the "Elsa" and "Surprised" attributes.

Reference-Based Style Editing. In our WarpGAN, the latent code plays a crucial role in controlling the inpainting process of SVINet. To more explicitly analyze the influence of the latent code, we

Figure 8: **Image attribute editing results** obtained by our method. The edited attributes include "Elsa", "Anger", "Old", "Young", and "Surprised".

Figure 9: **Reference-based style editing.** Each row represents the editing results of the same source image corresponding to different reference images, where the source and reference images are identical along the diagonal.

perform experiments by replacing the latent code of the input image during the inpainting process. Specifically, for the source image $\mathbf{I}_s$ with the camera pose c_s and the latent code w_s^+, we replace them with the camera pose c_r and the latent code w_r^+ of the reference image $\mathbf{I}_r$ during inpainting, thereby achieving simultaneous editing of view and style. The results are given in Fig. 9.

For our SVINet, w^+ modulates the convolutions in both N_I and N_D, where N_I processes feature maps at a resolution of 64×64, and N_D processes feature maps at resolutions ranging from 64×64 to 512×512. According to the characteristics of StyleGAN [20, 21], the latent code corresponding to feature maps at resolutions of 64×64 and above primarily controls the detailed features of the image, such as the color scheme and microstructure. From Fig. 9, we observe that the main changes are in the skin tone and hair color of the face.

Qualitative Evaluation in the Cat Domain. To further validate the generalization capability of our method, we evaluate it in the cat domain. Specifically, we use the AFHQ-CAT dataset [9] for training and evaluation. Following e4e [38], we use a ResNet50 network [16] trained with MOCOv2 [7] instead of the pre-trained ArcFace network [12] to compute the identity loss in the non-facial domains during training. As shown in Fig. 10, our method can generalize well to the cat domain and perform novel view synthesis as well as attribute editing.

Figure 10: **Novel view synthesis and attribute editing** on cat faces by our method. We visualize the novel view synthesis results of WarGAN and WarpGAN-Opt, as well as the editing results of the attributes "Color" and "Small Eyes".