Dual Encoder GAN Inversion for High-Fidelity 3D Head Reconstruction from Single Images

Bahri Batuhan Bilecen* **Ahmet Berke Gokmen*** **Aysegul Dundar**
Bilkent University, Department of Computer Engineering, Ankara, Türkiye
{batuhan.bilecen@, berke.gokmen@ug, adundar@cs}.bilkent.edu.tr

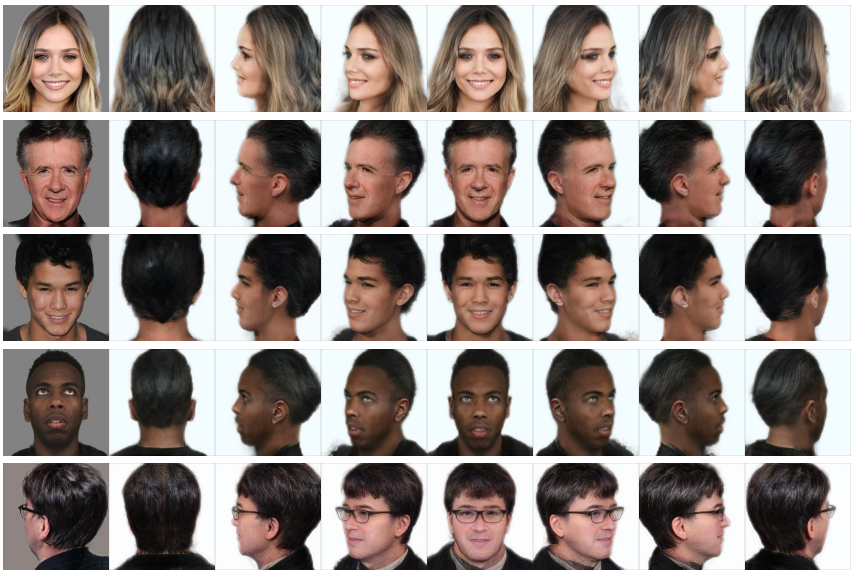

Figure 1: From a single input image (first column), our framework reconstructs 3D representation by inverting images into PanoHead's latent space, which can be viewed in a 360-degree perspective.

Abstract

3D GAN inversion aims to project a single image into the latent space of a 3D Generative Adversarial Network (GAN), thereby achieving 3D geometry reconstruction. While there exist encoders that achieve good results in 3D GAN inversion, they are predominantly built on EG3D, which specializes in synthesizing near-frontal views and is limiting in synthesizing comprehensive 3D scenes from diverse viewpoints. In contrast to existing approaches, we propose a novel framework built on PanoHead, which excels in synthesizing images from a 360-degree perspective. To achieve realistic 3D modeling of the input image, we introduce a *dual encoder system* tailored for high-fidelity reconstruction and realistic generation from different viewpoints. Accompanying this, we propose a *stitching framework on the triplane domain* to get the best predictions from both. To achieve seamless stitching, both encoders must output consistent results despite being specialized for different tasks. For this reason, we carefully train these encoders using specialized losses, including an adversarial loss based on our novel *occlusion-aware triplane discriminator*. Experiments reveal that our approach surpasses the existing encoder training methods qualitatively and quantitatively. Please visit the **project page**.

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2772

1 Introduction

In the realm of generative models, 2D GANs have gained renown for their remarkable ability to achieve striking realism through adversarial training, effectively capturing intricate details and textures to produce visually convincing images, especially on face images [18, 19]. However, their inherent limitation lies in their lack of depth perception, which restricts their applicability in three-dimensional contexts. In contrast, 3D GANs mark a groundbreaking advancement by seamlessly integrating Neural Radiance Fields (NeRF) into their architecture [13, 9, 14, 5]. This integration not only allows them to match the realism of their 2D counterparts but also ensures consistency in three-dimensional geometry. While extensive studies have focused on 2D GAN inversion [42, 29, 31, 4, 33, 28], recent efforts have seen the proposal of inversion methods tailored for 3D GANs [39, 7, 21].

2D GAN inversion techniques focus on projecting the images into the GAN's natural latent space to enhance editability and achieve high fidelity to the input image; however, inverting 3D GAN models presents additional challenges. This process requires accurate 3D reconstruction, which means realistic filling of invisible regions, ensuring coherence and completeness in the resulting three-dimensional scenes. Recently, inversion methods for 3D GANs have been developed, firstly, optimization-based and then encoder-based methods. Optimization-based methods [20, 35, 38] employ reconstruction losses to invert images into a latent code specific to the given view. Furthermore, network parameters are optimized by generating pseudo-multi-view images from the optimized latent codes to enhance detail preservation. Such optimization is required for each inference image; it is time-consuming and requires GPUs with large memory. Therefore, researchers focus on encoder-based methods [39, 7, 21]. While successful inversions are achieved with these methods, they rely on EG3D [9] framework [39, 7, 21], which is constrained to synthesizing near-frontal views. However, our work utilizes PanoHead [5], a method capable of rendering full-head image synthesis, enabling a comprehensive 360-degree perspective. This advancement introduces additional challenges, particularly concerning the invisibility of many parts in the input image. Despite this, the inversion model is expected to reasonably predict and reconstruct these occluded regions to ensure high-quality 3D reconstruction. Our experiments demonstrate that extending the methods proposed for EG3D is ineffective.

When projecting images onto PanoHead's latent space, we observe a trade-off between achieving high-fidelity reconstruction of the input image and generating realistic representations of the invisible parts of the head. Some models can perfectly reconstruct the image from a given view but produce unrealistic outputs when the camera parameters change. Conversely, other models generate realistic representations under varying camera parameters but fail to achieve high-fidelity reconstruction of the input image. To achieve high-fidelity reconstruction of the input image and realistic representations of the invisible parts of the head simultaneously, we train a **dual encoder**. One encoder specializes in reconstructing the given view, while the other focuses on generating high-quality invisible views. We propose stitching the triplane domain generations to produce the final result. This approach combines the outputs from both encoders to achieve both high-fidelity reconstructions of the given view and high-quality representations of the invisible parts of the head. To achieve seamless stitching, both encoders must output consistent results despite being specialized for different tasks. For this reason, we carefully train these encoders using specialized losses, including an adversarial loss based on our novel **occlusion-aware triplane discriminator**. This ensures that both encoders learn to produce consistent and complementary outputs, enabling seamless stitching of generations for the final result. Our contributions are as follows:

- To achieve high fidelity to the input and realistic generations for different camera views, we train dual encoders and introduce a stitching pipeline that combines the best predictions from both encoders for visible and invisible regions.
- We propose a novel occlusion-aware discriminator that enhances both fidelity and realism.
- We conduct extensive experiments to show the effectiveness of our framework. Quantitative and qualitative results show the superiority of our method compared to the state-of-the-art. Some visual results can be seen in Fig. 1.

2 Related works

3D Generative Models. Generative Adversarial Networks (GANs), coupled with differentiable renderers, have achieved significant strides in generating 3D-aware multi-view consistent images.

While early efforts, such as HoloGAN [24], operating on voxel representations, subsequent works shifted towards mesh representations [27, 15], and the latest advancements are built around implicit representations [10, 14, 26, 25]. Among implicit representations, triplane representations have emerged as a popular choice due to their computational efficiency and the high-quality outputs [9, 13, 5]. The architectures of works like EG3D [9] and PanoHead [5] bear resemblance to the structure of StyleGAN2 [19]. They consist of mapping and synthesis networks, generating triplanes which are subsequently projected to a 2D image through volumetric rendering operations akin to those used in NeRF [23]. While EG3D is trained on the FFHQ [18] dataset with limited angle diversity, PanoHead achieves a 360-degree perspective in face generation thanks to their dataset selection and model improvements. In our work, we delve into PanoHead's latent space and construct our inversion encoder based on PanoHead.

GAN Inversion. In recent years, GAN inversion, particularly in the context of StyleGAN, has garnered significant attention due to its extensive editing capabilities. The primary objective of these studies is to embed an image into StyleGAN's latent space, enabling subsequent modifications. Initially, this was approached through latent optimization, where latent codes were iteratively adjusted using back-propagation to minimize the reconstruction loss between the generated and target images [11, 1, 2, 19, 30]. For 3D-aware GAN model inversions, supplementary heuristics have been introduced into the optimization process. These include considerations like facial symmetry [38] and multi-view optimization strategies [35]. However, such methods are computationally intensive as they require optimizing each image's latent codes. Moreover, while minimizing the reconstruction loss can yield visually similar results, it does not guarantee that the image resides within the natural latent space of GANs. This distinction is crucial for effective image editing. Without aligning with StyleGAN's inherent latent space, reconstructed images may not respond correctly to editing techniques, thus limiting their practical utility. This consideration also extends to 3D-aware GAN inversion methods, where encoding geometric information is paramount. Even if an input image can be faithfully reconstructed, its realism may falter when observed from alternative viewpoints, emphasizing the importance of aligning with the GAN's native latent space.

To enhance efficiency, image encoders have been specifically trained for the inversion task, initially targeting StyleGAN [42, 29, 31, 4, 28, 36, 37], and more recently for EG3D [39, 7, 21]. These specialized encoders capitalize on insights gained from training datasets to swiftly project images into latent spaces. Moreover, they can be trained with diverse objectives beyond mere image reconstruction. For instance, some employ discriminators to compare generated and real images and latent space discriminators to ensure inversion aligns with the GAN's natural latent space. As a result, these methods generally offer faster inversion processes. This study focuses on 3D-GAN inversion, specifically targeting PanoHead [5]. The task poses significant challenges due to PanoHead's ability to capture a comprehensive 360-degree perspective, necessitating the prediction of a substantial portion of invisible elements by inversion encoders. Our experiments demonstrate that models trained for EG3D are ineffective in this context.

3 Method

3.1 Overview of PanoHead

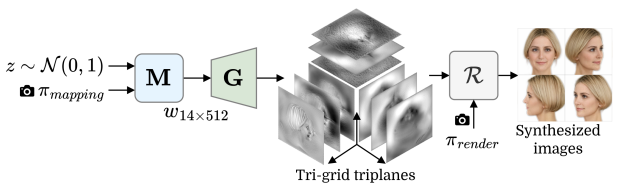

Figure 2: Overall architecture of PanoHead.

The overall architecture of PanoHead is given in Fig. 2. Resemblant to EG3D, PanoHead utilizes a mapping network that takes a random vector z and the camera conditioning π_{mapping}. After z is mapped to a w, StyleGAN-based backbone **G** generates a tri-grid triplane. Unlike EG3D, PanoHead's triplanes have 3 times the number of channels in comparison, hence the name tri-grid. This approach is stated to ease 360-degree synthesis. The resultant triplane is then rendered via a volumetric neural renderer $\mathcal{R}$ with pose π_{render} and super-resolved to yield a synthesized image.

https://doi.org/10.52202/079017-2772

3.2 Training an encoder

This section introduces the general pipeline of the encoders employed in our dual-encoder framework. Each encoder takes an input image $\mathbf{I}$ and predicts the latent code w^+, which is then passed to the generator to produce triplane features. These synthesized features are then fed into the renderer to generate a 2D image with a specified camera parameter π_{cam}, as described by Eq. (1):

$$\begin{aligned} w^+ &= \mathbf{E_1}(\mathbf{I}) \\ \mathbf{I^{sv}_{out}} &= \mathcal{R}(\mathbf{G}(w^+), \pi_{cam}) \end{aligned} \tag{1}$$

Here, $\mathbf{I^{sv}_{out}}$ denotes the output rendering for the same view as the input, and $\mathbf{E_1}$ represents the encoder. While the $\mathcal{W}^+$ space allows for leveraging priors embedded in the generator, its limited expressive power in reconstructing image details has been noted due to the information bottleneck of its 14×512 dimensions. To address this limitation, both 2D inversion techniques [33, 28] and 3D GAN inversion methods [7] permit higher-rate features to pass to the generators, facilitating the capture of fine details. In 3D GAN inversion methods, these higher-rate features are encoded through a smaller second network and transmitted to the triplane features. We adopt a similar approach in our encoders. We refer to the final output as $\mathbf{I^{sv}_{final}}$. Further details of the architecture are provided in the Appendix.

The primary challenge in this setting arises from establishing appropriate training objective losses, as our training dataset consists solely of single images, providing ground truth only for the rendered image from the same view as the input. For these output and ground-truth pairs, we set the usual reconstruction losses, namely, LPIPS perceptual loss [41], $\mathcal{L}_2$ reconstruction loss (MSE), and ArcFace [12] based identity loss as given in Eq. (2):

$$\arg\min_{\mathbf{E_1}} \mathcal{L}_{\text{LPIPS}}(\mathbf{I^{sv}_{final}}, \mathbf{I}) + \mathcal{L}_2(\mathbf{I^{sv}_{final}}, \mathbf{I}) + \mathcal{L}_{\text{identity}}(\mathbf{I^{sv}_{final}}, \mathbf{I}) \tag{2}$$

While models trained with the objective given in Eq. (2) learn to reconstruct a given view, they often struggle to generalize and produce realistic features from other camera views. Consequently, while our first encoder is trained with the objective in Eq. (2), we design an adversarial-based loss objective for our second encoder. This second encoder generates realistic predictions for invisible views, as explained in Section 3.3.

3.3 Occlusion-aware triplane discriminator

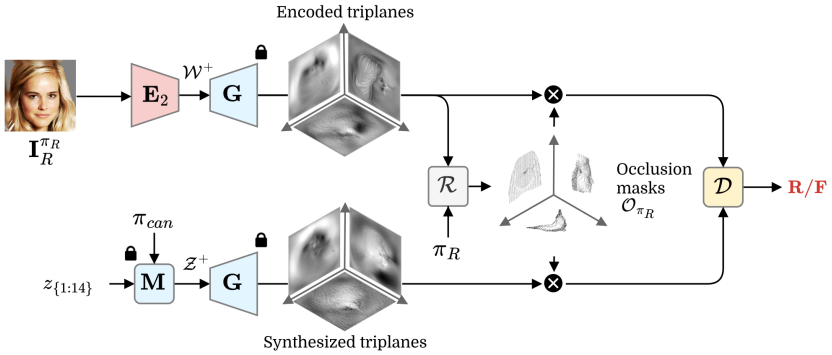

Figure 3: Our training methodology for the triplane discriminator involves generating real samples by sampling latent vectors $\mathcal{Z}^+$ and producing in-domain triplanes using PanoHead. Fake samples are generated from encoded images. Despite the effectiveness of adversarial loss in enhancing reconstructions, challenges may persist in achieving high fidelity to the input due to the origin of real samples from the generator $\mathbf{G}$. To address this, we propose an **occlusion-aware discriminator** $\mathcal{D}$, trained exclusively with features from occluded pixels. This ensures that visible regions, such as frontal views π_R, have reduced influence during the training of $\mathcal{D}$.

To achieve a realistic reconstruction of the 3D model, represented in a triplane structure, it is essential to guide the encoder for visible views and overall coherence. Since we lack one-to-one ground truth

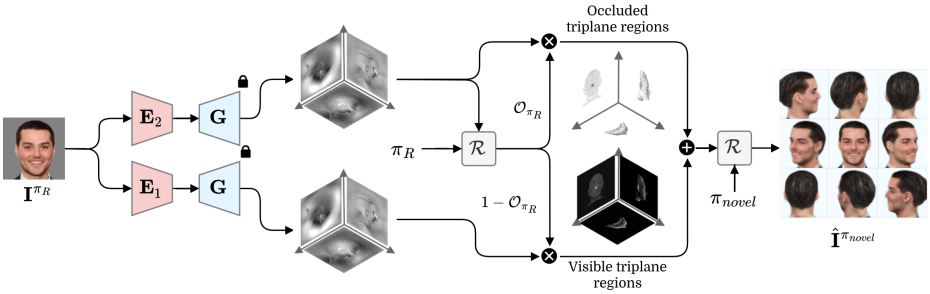

Figure 4: The inference pipeline with dual encoders for full 3D head reconstruction. Given a face portrait with pose π_R, we can perform 360-degree rendering from any given pose π_{novel}.

to guide the triplane structure, we experiment with various setups incorporating adversarial losses. A naive approach to utilize adversarial loss would be to render estimated triplanes from other views and assess the realism of these 2D images using a discriminator. However, our experiments observe that this setup hinders the model's ability to learn high fidelity to the input image, as will be further detailed in Section 5.2. Moreover, randomly rendering different views can only guide limited parts of the triplane structure rather than the overall.

To overcome this limitation and avoid the computational burden of rendering unnecessary views, we explore the possibility of training a discriminator in the triplane domain. In our training process for the triplane discriminator, we follow a procedure where we sample latent vectors $\mathcal{Z}^+$ and generate in-domain triplanes using PanoHead, serving as our real samples. Meanwhile, the fake samples are triplanes generated from encoded images, as depicted in Fig. 3. Despite the observed improvement in reconstructions facilitated by this adversarial loss, we note a persistent challenge hindering the network's ability to achieve high fidelity to the input. This discrepancy may stem from the real samples originating from the generator, lacking the detailed feature characteristic of real-world images. Therefore, this may lead the encoder to omit to encode realistic facial details if they are absent in the synthesized samples. We propose our **occlusion-aware discriminator** to overcome this limitation. This discriminator is exclusively trained with features corresponding to occluded pixels. This approach ensures that triplane features associated with visible regions, such as a frontal face, are not utilized for discriminator training. Additionally, we introduce a masking mechanism for synthesized triplanes to mitigate any distribution mismatch arising between encoded and synthesized triplanes. This masking process contributes to aligning the distributions of real and fake samples, further enhancing the coherence of the training dynamics.

We find the set of visible points based on the depth map of the given view via inverse rendering. Specifically, the occlusion mask $\mathcal{O}_{\pi_R}$ is estimated by Eq. (3):

$$\mathcal{O}_{\pi_R} = \mathbb{R}^3 \setminus \{\mathbf{p}[x, y, z] \ : \ \pi_R \mathcal{D} \mathcal{K}^{-1} \mathbf{I}[u, v, 1]^T\} \tag{3}$$

where $\mathbf{p}[x, y, z]$ is the triplane coordinates, π_R is the extrinsic camera parameters of the input view, $\mathcal{D}$ is the depth map from the input view, $\mathcal{K}$ is the intrinsic camera parameters, $\mathbf{I}[u, v, 1]$ are the homogeneous coordinates of the input image $\mathbf{I}$ rendered from the input view. More clearly, from the current camera pose, we map back to the depth values to obtain the mask of visible regions (1-$\mathcal{O}_{\pi_R}$). Then, we invert the visible region mask to obtain $\mathcal{O}_{\pi_R}$. The utilization of occlusion masks has been previously investigated in 3D methodologies, albeit in different contexts. For instance, they have been used in generating pseudo-ground truth images to facilitate optimization-based 3D reconstruction [38] and integrated into passing high-rate residual features to the triplane [39]. However, it is the first time used in the discriminator. This allows for a selective focus on regions where the encoder may encounter challenges in faithfully replicating realism.

Compliant with recent advancements in adversarial training, we follow WGAN loss [6] for $\mathcal{L}_{\text{adv}}$ in Eq. (4), where $\mathbf{T}^{\mathbf{sv}}_{\mathbf{final}}$ and $\mathbf{T}_{\mathbf{synth}}$ are encoded and $\mathcal{Z}^+$ synthesized triplanes, respectively. Details are given in the Appendix.

$$\arg\min_{\mathbf{E_2}} \max_{\mathcal{D}} \mathcal{L}_{\text{adv}}(\mathcal{O}_{\pi_R} \mathbf{T}^{\mathbf{sv}}_{\mathbf{final}}, \mathcal{O}_{\pi_R} \mathbf{T}_{\mathbf{synth}}) \tag{4}$$

https://doi.org/10.52202/079017-2772

3.4 Dual encoder pipeline

Figure 5: Visual results of Encoder 1, Encoder 2, and Dual encoders for the given input images in the first and sixth columns.

In our approach, we train two encoders: the first, as outlined in Section 3.2, and the second, augmented with an additional adversarial loss detailed in Section 3.3. While the initial encoder excels at reconstructing high-fidelity facial images from the input, it often produces unrealistic results for other viewpoints, as depicted in Fig. 5. Conversely, the second encoder yields better overall outcomes, albeit with slightly diminished fidelity to the input face.

Our aim is to devise a dual-encoder pipeline that harnesses the strengths of both encoded features. To achieve this, we leverage the occlusion masks derived in Section 3.3, as illustrated in Fig. 4. By combining the visible portions from Encoder 1 and the occluded segments from Encoder 2, we generate our final output, as demonstrated in the last row of Fig. 5.

While each encoder contributes partially to the ultimate feature, achieving seamless integration necessitates consistency in the output of both encoders despite their distinct specializations. For instance, if Encoder 1 flawlessly renders a given view of the face but fails to capture the correct geometry, artifacts may arise in the combined result. Thus, it remains imperative to train both encoders comprehensively to ensure an overall high-quality outcome.

4 Experimental Setup

4.1 Training

We combined images of FFHQ [18] and LPFF [34] and split it for training (∼140k) and validation (∼14k). CelebA-HQ [17] and multi-view MEAD [32] are employed for additional evaluation. We removed face portrait backgrounds for training and evaluation datasets, applied camera and image mirroring during training, and performed pose rebalancing proposed in [34] as data augmentation. We utilized the same dataset to train competitive methods for fair evaluation. The models are trained for 500k iterations with a batch size of 3 on a single RTX 4090 GPU. The learning rate is $1e^{-4}$ for both encoders and the occlusion-aware discriminator. Ranger is utilized as the optimizer, which is a combination of Rectified Adam [22] with Lookahead [40].

4.2 Baselines

The baseline models are provided in Table 1. We note that no encoder pipelines are aimed for full 360-degree head reconstruction. We train the models with the author's released code to invert images into PanoHead's latent space.

4.3 Evaluation metrics

We report $\mathcal{L}_2$, LPIPS [41] and ID [12] scores for original-view reconstruction, which measure the fidelity to the input image. For the novel-view quality, we measure Fréchet inception distance (FID) [16]. Since our validation datasets have limited angle variance, we measure the distance between 1k randomly synthesized and 1k encoded real-life image distributions. The images are rendered from varying yaw angles, covering the 360-degree range to include occluded regions. We

also utilize the multi-view image dataset MEAD dataset. Specifically, we fed front MEAD images (0° yaw) to all methods, rendered them from novel views of MEAD (from 60° to 0° yaw), and compare them with their corresponding ground truths. This allows us to report LPIPS and ID metrics alongside the FID metric for the novel views.

5 Results

5.1 Comparisons with state-of-the-art

Table 1: Quantitative scores on various test sets.

Category	Method	FFHQ + LPFF				CelebAHQ				Time
		$\mathcal{L}_2 \downarrow$	LPIPS $\downarrow$	ID $\uparrow$	FID $\downarrow$	$\mathcal{L}_2 \downarrow$	LPIPS $\downarrow$	ID $\uparrow$	FID $\downarrow$	sec $\downarrow$
Optimization	$\mathcal{W}^+$ optim. [2]	0.035	0.17	0.57	65.30	0.049	0.19	0.49	63.57	49.19
	PTI [30]	0.019	0.11	0.89	59.40	0.033	0.13	0.86	57.85	86.52
Encoder	pSp[29]	0.028	0.15	0.77	90.52	0.033	0.17	0.70	88.20	0.08
	e4e[31]	0.064	0.24	0.52	82.99	0.080	0.26	0.42	84.70	0.05
	TriplaneNetv2[7]	0.017	0.10	0.86	98.97	0.020	0.11	0.80	99.02	0.17
	GOAE[39]	0.015	0.09	0.87	168.96	0.017	0.10	0.86	173.64	0.15
	Ours	0.017	0.10	0.87	65.44	0.021	0.12	0.84	62.58	0.37

Table 2: Quantitative scores on multi-view MEAD dataset.

Category	Method	LPIPS $\downarrow$		ID $\uparrow$		FID $\downarrow$	
		$\pm 60^\circ$	$\pm 30^\circ$	$\pm 60^\circ$	$\pm 30^\circ$	$\pm 60^\circ$	$\pm 30^\circ$
Optim.	$\mathcal{W}^+$opt.	0.249	0.200	0.570	0.558	48.473	43.875
	PTI	0.346	0.262	0.515	0.548	66.399	53.977
Encoder	pSp	0.245	0.189	0.640	0.650	49.364	48.098
	e4e	0.318	0.265	0.544	0.462	67.399	70.854
	Tpn.v2	0.248	0.192	0.658	0.663	48.937	46.493
	GOAE	0.296	0.249	0.654	0.660	87.644	92.758
	Ours	0.223	0.178	0.706	0.726	47.207	43.822

Table 1 provides quantitative comparisons against state-of-the-art optimization and encoder-based methods. GOAE achieves significantly better same-view reconstruction scores (L2, LPIPS, and ID), however, shows much worse FID scores, indicating their inability to produce realistic views. While TriplaneNetv2 achieves similar same-view reconstruction scores as our method, its FID score is also significantly worse. Overall, the pSp and e4e methods perform worse than ours in all metrics. PTI achieves similar results to our method but takes $\times 250$ longer and requires a GPU with large memory.

We extend the quantitative analyses to multi-view with the MEAD dataset. Specifically, we feed front MEAD images (0° yaw) to all methods, rendered them from novel views of MEAD (from 60° to 0° yaw), and compare them with their corresponding ground truths. Table 2 reveals that our method significantly improves over compared methods especially in LPIPS and ID metrics.

Qualitative results are shown in Fig. 8 and Fig. 6. The competing methods produce unrealistic outputs when viewed from angles other than the input view. Among these, PTI achieves good front and side views but fails to generate realistic hair from the back. Our method achieves the best results overall.

We also include mesh comparisons in Fig. 9. Ours is better than the most recent encoder-based method [7] and generally performs well compared to PTI. Note that PTI mostly generates smoother meshes (row 3) but can sometimes struggle depending on the input sample (row 6).

5.2 Ablation study

Table 3: Ablation on occlusion-aware discriminator $\mathcal{D}$.

	LPIPS $\downarrow$	ID $\uparrow$	FID $\downarrow$
No $\mathcal{D}$	0.10	0.87	89.50
$\mathcal{D}$ w image domain	0.17	0.67	72.86
$\mathcal{D}$ w/o triplane occ.	0.15	0.70	66.24
$\mathcal{D}$ **w/ triplane occ.**	0.14	0.75	64.02

In Table 3 and Fig. 7, we present an ablation study demonstrating the effectiveness of our occlusion-aware triplane discriminator quantitatively and qualitatively. The first row of results shows that not using any discriminator achieves good reconstruction of the given view, as indicated by LPIPS and ID scores. This is also visible in the first-row in Fig. 7. However, this approach fails to generalize to novel views, as evidenced by the FID score and visual results.

https://doi.org/10.52202/079017-2772

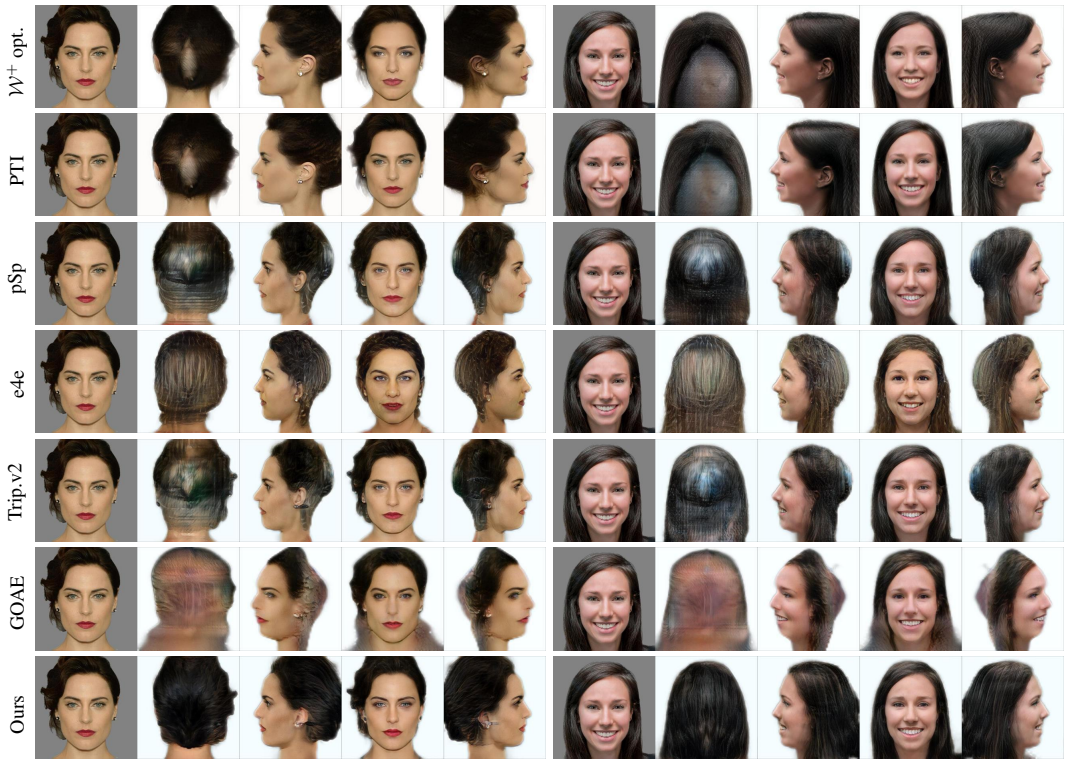

Figure 6: Comparisons of ours and competing methods.

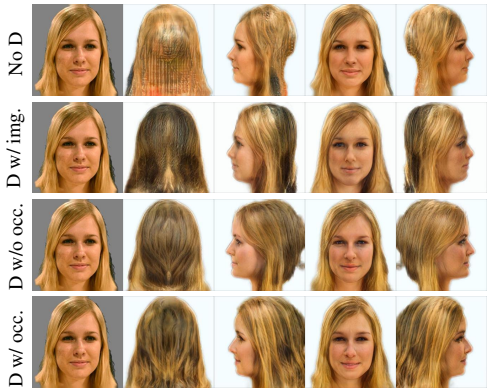

Figure 7: Qualitative results of ablation on occlusion-aware discriminator $\mathcal{D}$.

Table 4: Ablation on training data and latent space.

Train data	Proj.	LPIPS ↓	ID ↑	FID ↓
Real img.	$\mathcal{Z}$	0.29	0.31	45.79
Real img.	$\mathcal{Z}^+$	0.22	0.60	76.54
Real img.	$\mathcal{W}^+$	0.10	0.86	98.97
$\mathcal{Z}^+$ gens.	$\mathcal{W}^+$	0.27	0.25	46.93
Real imgs. + $\mathcal{Z}^+$	$\mathcal{W}^+$	0.10	0.87	89.50

On the other hand, training the model with an additional adversarial objective that operates on novel images generated using randomly sampled camera parameters improves the FID score but significantly harms the fidelity of the input image. Training a discriminator in the triplane domain and applying adversarial losses from this domain improves overall scores compared to training the discriminator in the 2D image domain. However, as seen in Fig. 7 (third row), the face still lacks high fidelity to the input, and other views are unrealistic. Lastly, using the occlusion-aware triplane discriminator improves identity fidelity and FID scores. The hair looks more natural, similar to the ones generated by the model when sampled from z.

In our framework, we chose to embed images into the $\mathcal{W}^+$ space. Table 4 presents an ablation study that explores utilizing different projection spaces and various combinations of training data. Training an encoder to project images to the $\mathcal{Z}$ or $\mathcal{Z}^+$ space, where $\mathcal{Z}$ is sampled 14 times, results in better FID scores. However, this comes at the cost of high-fidelity reconstruction. Similarly, transitioning to a less constrained $\mathcal{W}^+$ space enhances fidelity to the input but worsens the FID score. Addressing this challenge necessitates additional measures, such as the proposed dual encoder setup with the occlusion-aware discriminator objective. It is important to note that the distinction between the $\mathcal{Z}^+$ and $\mathcal{W}^+$ space arises from the camera parameters incorporated into the mapping network. While $\mathcal{Z}^+$ employs various samples of $\mathcal{Z}$, it adheres to the same set of camera parameters assigned to the mapping network. In contrast, the $\mathcal{W}^+$ space does not impose

Figure 8: Left to right: input ($0°$), reconstruction ($0°$), GT target ($\pm 60°$), and render on $\pm 60°$ using the reconstruction triplanes of $0°$ on MEAD dataset.

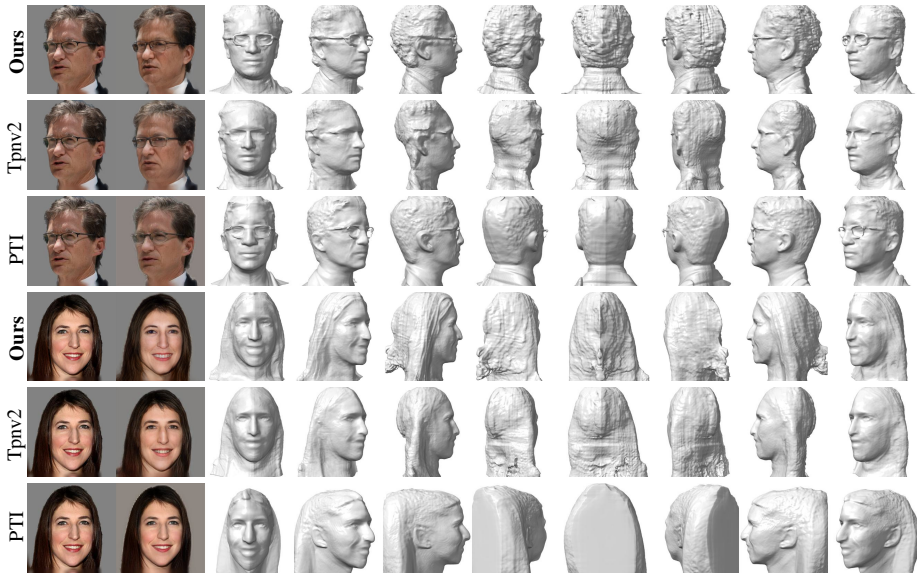

Figure 9: Inputs (first), reconstructions (second), and $360°$ mesh renders (rest) of our method.

such constraints during encoding. Given that the real image dataset we use primarily consists of limited camera poses, typically front-view faces, we investigate training the encoder with synthetically generated images from PanoHead. However, solely utilizing synthetic images generated from samples of $\mathcal{Z}^+$ to introduce more diversity compared to $\mathcal{Z}$ leads to poor performance on real image validation sets regarding reconstruction quality. When combining synthetic and real images, we observe an improvement compared to using them individually.

https://doi.org/10.52202/079017-2772

Figure 10: Hair edits from source image (first) to destination image (second) and 360° renders (rest).

Table 5: Ablation on dual-encoder.

Method	LPIPS ↓	ID ↑	FID ↓
$\mathbf{E_1}$	0.10	0.87	89.50
$\mathbf{E_2}$	0.14	0.75	64.02
Dual	0.10	0.87	65.44

Lastly, we show the results of using the dual encoder in Table 5. The visual results were previously presented in Fig. 5. The dual mechanism leverages the strengths of both encoders, achieving the same LPIPS and ID scores as Encoder 1 while also producing FID scores very similar to those of Encoder 2.

5.3 Editing application

We follow the reference-based editing in [8] in our pipeline. This method encodes input images, and edits are performed in the triplane space. This approach utilizes the fact that triplanes have a canonical space, allowing for the transfer of local parts from one triplane to another. Fig. 10 demonstrates a successful transfer of hairstyle from a reference image to the target human in 3D. Another advantage of encoder-based models over the optimization ones is the feasibility of such applications. For example, this would not be possible with PTI since the generator is fine-tuned for each sample, preventing the copying of features from one image to another in the encoded feature space.

6 Conclusion and Broader Impacts

Figure 11: Example failure cases. Inputs (first), reconstructions (second), and novel views.

In summary, this study introduces a 3D GAN inversion framework that projects single images into the latent space of a 3D GAN for accurate 3D geometry reconstruction. While prior encoders excel at synthesizing near-frontal views, they struggle with diverse 3D scenes, motivating our exploration of alternatives. Using PanoHead's 360-degree synthesis, we developed a **dual encoder system** for high-fidelity reconstruction and realistic multi-view generation. A stitching mechanism in the triplane domain ensures optimal predictions from both encoders. With specialized losses, including an **occlusion-aware triplane discriminator**, our framework achieves superior qualitative and quantitative performance over existing methods.

Broader Impacts. Our framework has the potential to revolutionize the movie industry, AR, and VR, enabling applications like animating portraits and creating realistic game environments. However, it raises ethical concerns, particularly the risk of "deep fakes". We stress the need for safeguards to ensure the ethical use of this technology.

Limitations. We acknowledge that there is room for improvements in the fidelity of images, the realism and 3D-consistency of generations (see Fig. 11, row 2), and the smoothness of the meshes (see Fig. 9). Since the projection is made onto the latent space of PanoHead, our method may not handle out-of-domain or tail samples well (such as images with high-frequency details or accessories). For instance, our method struggles with hats, as demonstrated in the first row of Fig. 11. We recognize that, in certain cases, the artifacts are visible in the back middle of the head and are more noticeable in the mesh rendering as shown in Fig. 9. Additional research is required.

Acknowledgements. This work was supported by the BAGEP Award of the Science Academy. We acknowledge EuroHPC Joint Undertaking for awarding the project ID EHPC-AI-2024A02-031 access to Leonardo at CINECA, Italy.

References

[1] Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4432–4441 (2019)

[2] Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8296–8305 (2020)

[3] Alaluf, Y., Patashnik, O., Cohen-Or, D.: Restyle: A residual-based stylegan encoder via iterative refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6711–6720 (2021)

[4] Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18511–18521 (2022)

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

[6] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan (2017)

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

[8] Bilecen, B.B., Yalin, Y., Yu, N., Dundar, A.: Reference-based 3d-aware image editing with triplanes. arXiv preprint arXiv:2404.03632 (2024)

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

[10] Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: 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. pp. 5799–5809 (2021)

[11] Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. IEEE transactions on neural networks and learning systems **30**(7), 1967–1974 (2018)

[12] Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)

[13] Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. Advances In Neural Information Processing Systems **35**, 31841–31854 (2022)

[14] Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. arXiv preprint arXiv:2110.08985 (2021)

[15] Henderson, P., Tsiminaki, V., Lampert, C.H.: Leveraging 2d data to learn textured 3d mesh generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7498–7507 (2020)

[16] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)

[17] Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)

 https://doi.org/10.52202/079017-2772

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

[19] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020)

[20] Ko, J., Cho, K., Choi, D., Ryoo, K., Kim, S.: 3d gan inversion with pose optimization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2967–2976 (2023)

[21] Li, X., De Mello, S., Liu, S., Nagano, K., Iqbal, U., Kautz, J.: Generalizable one-shot 3d neural head avatar. Advances in Neural Information Processing Systems **36** (2024)

[22] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the variance of the adaptive learning rate and beyond. In: Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020) (April 2020)

[23] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)

[24] Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7588–7597 (2019)

[25] Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11453–11464 (2021)

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

[27] Pavllo, D., Spinks, G., Hofmann, T., Moens, M.F., Lucchi, A.: Convolutional generation of textured 3d meshes. Advances in Neural Information Processing Systems **33**, 870–882 (2020)

[28] Pehlivan, H., Dalva, Y., Dundar, A.: Styleres: Transforming the residuals for real image editing with stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1828–1837 (2023)

[29] Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2287–2296 (2021)

[30] Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. ACM Transactions on Graphics (TOG) **42**(1), 1–13 (2022)

[31] Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG) **40**(4), 1–14 (2021)

[32] Wang, K., Wu, Q., Song, L., Yang, Z., Wu, W., Qian, C., He, R., Qiao, Y., Loy, C.C.: Mead: A large-scale audio-visual dataset for emotional talking-face generation. In: ECCV (August 2020)

[33] Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: High-fidelity gan inversion for image attribute editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11379–11388 (2022)

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

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

[36] Yildirim, A.B., Pehlivan, H., Bilecen, B.B., Dundar, A.: Diverse inpainting and editing with gan inversion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23120–23130 (2023)

[37] Yildirim, A.B., Pehlivan, H., Dundar, A.: Warping the residuals for image editing with stylegan. arXiv preprint arXiv:2312.11422 (2023)

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

[39] Yuan, Z., Zhu, Y., Li, Y., Liu, H., Yuan, C.: 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 (2023)

[40] Zhang, M., Lucas, J., Ba, J., Hinton, G.E.: Lookahead optimizer: k steps forward, 1 step back. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)

[41] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)

[42] Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain gan inversion for real image editing. In: European conference on computer vision. pp. 592–608. Springer (2020)

 https://doi.org/10.52202/079017-2772

A Appendix

A.1 Architecture details

Conv2D(288,64,3,1,1) LeakyReLU(0.2)
Conv2D(64,128,3,1,1) BatchNorm2D(128) LeakyReLU(0.2)
Conv2D(128,256,3,1,1) BatchNorm2D(256) LeakyReLU(0.2)
Conv2D(256,512,3,1,1) BatchNorm2D(512) LeakyReLU(0.2)
Conv2D(512,1,3,1,0) AvgPool(1)

Table 6: Architecture for discriminator. Conv2D parameters are: (input channels, output channels, kernel size, stride, padding), respectively. Bias terms are disabled.

Encoder 1 and 2. For our Encoder 1 and 2, we employ 2-stage encoding for $\mathcal{W}^+$ and high-rate $\mathcal{F}$ features, also seen in common with other style-based encoder methods [29, 3, 4, 39, 7]. We opted for the architecture in [29] for $\mathcal{W}^+$ stage (`GradualStyleEncoder`) and in [7] for $\mathcal{F}$ stage (`TriplanenetEncoder`), both for Encoder 1 and 2.

Ablation encoders. `GradualStyleEncoder` is used for the $\mathcal{Z}^+$ encoder, where resulting 14 latent vectors are later passed through the mapping network with truncation $\psi = 0.85$ and canonical front camera pose. However, for the $\mathcal{Z}$ encoder, a smaller variation of [29] (`BackboneEncoderUsingLastLayerIntoW`) is implemented. This choice is due to $\mathcal{Z}$ being less expressive compared to $\mathcal{Z}^+$ (1 dim vs. 14 dim) and hence a smaller encoder being sufficient. The same mapping network parameters for the $\mathcal{Z}^+$ case are also utilized for $\mathcal{Z}$ encoder outputs.

Occlusion-aware triplane discriminator. We follow a feedforward network approach with a channel bottleneck for our triplane discriminator $\mathcal{D}$ (Table 6). Noting that the occluded triplane dimensions are `[batch_size,3,96,256,256]`, we first add each depth slice to the channel dimension to get the input shape as `[batch_size,288,256,256]`. Output dimensions are `[batch_size,1]`. We do not utilize saturating functions such as sigmoid at the end since WGAN-based loss [6] is utilized.

Ablation image discriminator. For the back-view image discriminator used in ablations, we change the input channel number of the model in Table 6 from 288 to 3.

A.2 Training objectives and hyperparameters

The training objective for **Encoder 1** with parameters $\theta_{\mathbf{E}_1}$ is given in Eq. (5).

$$\begin{aligned} \mathbf{I}^{\mathbf{sv}} &= \mathcal{R}(\mathbf{G}(\mathbf{E_1}(\mathbf{I})), \pi) \\ \arg\min_{\theta_{\mathbf{E}_1}} \ \lambda_1 \mathcal{L}_{\text{LPIPS}}(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) + \lambda_2 \mathcal{L}_2(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) + \lambda_3 \mathcal{L}_{\text{identity}}(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) \end{aligned} \tag{5}$$

where $\mathbf{I}^{\mathbf{sv}}$ is the same-view reconstruction of $\mathbf{I}$ with pose π. Coefficients are set as $\lambda_1 = 0.8$, $\lambda_2 = 1.0$, $\lambda_3 = 0.5$.

The training objective for **Encoder 2** with parameters $\theta_{\mathbf{E}_2}$ is given in Eq. (6).

$$\begin{aligned} \mathbf{T}_{\text{enc}} &= \mathbf{G}(\mathbf{E_2}(\mathbf{I})) \\ \mathbf{I}^{\mathbf{sv}} &= \mathcal{R}(\mathbf{T}_{\text{enc}}, \pi) \\ \arg\min_{\theta_{\mathbf{E}_2}} \ \lambda_1 \mathcal{L}_{\text{LPIPS}}(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) + \lambda_2 \mathcal{L}_2(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) + \lambda_3 \mathcal{L}_{\text{identity}}(\mathbf{I}^{\mathbf{sv}}, \mathbf{I}) + \lambda_4 \mathcal{L}_{\text{adv}}(\mathcal{D}(\mathcal{O}_\pi \mathbf{T}_{\text{enc}})) \end{aligned} \tag{6}$$

where $\mathbf{T}_{\text{enc}}$ is the encoded triplane of image $\mathbf{I}$, $\mathcal{D}$ is the discriminator, $\mathcal{O}_\pi$ is the occlusion mask from the same view π. $\lambda_{1,2,3}$ are the same as in Eq. (5), and $\lambda_4 = 0.001$. $\mathcal{L}_{\text{adv}}$ in Eq. (6) is given in Eq. (7):

$$\mathcal{L}_{\text{adv}}(x) = \texttt{softplus}(-x) \tag{7}$$

where `softplus` is a smooth and differentiable approximation to ReLU.

The training objective for **discriminator** $\mathcal{D}$ with parameters $\theta_{\mathcal{D}}$ is given in Eq. (8).

$$
\begin{aligned}
\mathbf{T}_{\text{enc}} &= \mathbf{G}(\mathbf{E_2}(\mathbf{I})) \\
\mathbf{T}_{\text{synth}} &= \mathbf{G}(\mathbf{M}(z^+ \sim \mathcal{N}(0, \mathrm{I}_{14}), \pi_{\text{front}})) \\
\arg\min_{\theta_{\mathcal{D}}} \ & \lambda \mathcal{L}_{\text{adv}}(\mathcal{D}(\mathcal{O}_\pi \mathbf{T}_{\text{enc}}),\ \mathcal{D}(\mathcal{O}_\pi \mathbf{T}_{\text{synth}}))
\end{aligned}
\tag{8}
$$

where $\mathbf{T}_{\text{synth}}$ is the synthesised triplane from randomly sampled z^+, $\mathbf{M}$ is the mapping network, π_{front} is the canonical front pose. $\mathcal{L}_{\text{adv}}$ in Eq. (8) is given in Eq. (9):

$$
\mathcal{L}_{\text{adv}}(x, y) = \texttt{softplus}(-x) + \texttt{softplus}(y) \tag{9}
$$

λ is set as 0.5.

We jointly train Encoder 2 and discriminator $\mathcal{D}$ in a traditional adversarial fashion. We further employ R1-regularization to encourage L1-lipschitzness [6] to justify using `softplus`, where its weighting coefficient is 10 and is applied every 16 iterations.

A.3 Additional qualitative results

Our method can handle diverse ethnicities and challenging input views, demonstrated in Fig. 12. We also showcase additional visual results for competing methods in Figs. 13 to 20, ablation on discriminator in Fig. 21, ablation on dual-encoder structure in Figs. 22 to 24. The first columns are input images, the second columns are reconstructions from the input views, and the rest are renderings of models from novel views.

https://doi.org/10.52202/079017-2772

Figure 12: Inputs with diverse ethnicities and challenging views (first), reconstructions (second), and 360° renders (rest).

Figure 13: Comparisons of ours and competing methods.

Figure 14: Comparisons of ours and competing methods.

https://doi.org/10.52202/079017-2772

Figure 15: Comparisons of ours and competing methods.

Figure 16: Comparisons of ours and competing methods.

https://doi.org/10.52202/079017-2772

Figure 17: Comparisons of ours and competing methods.

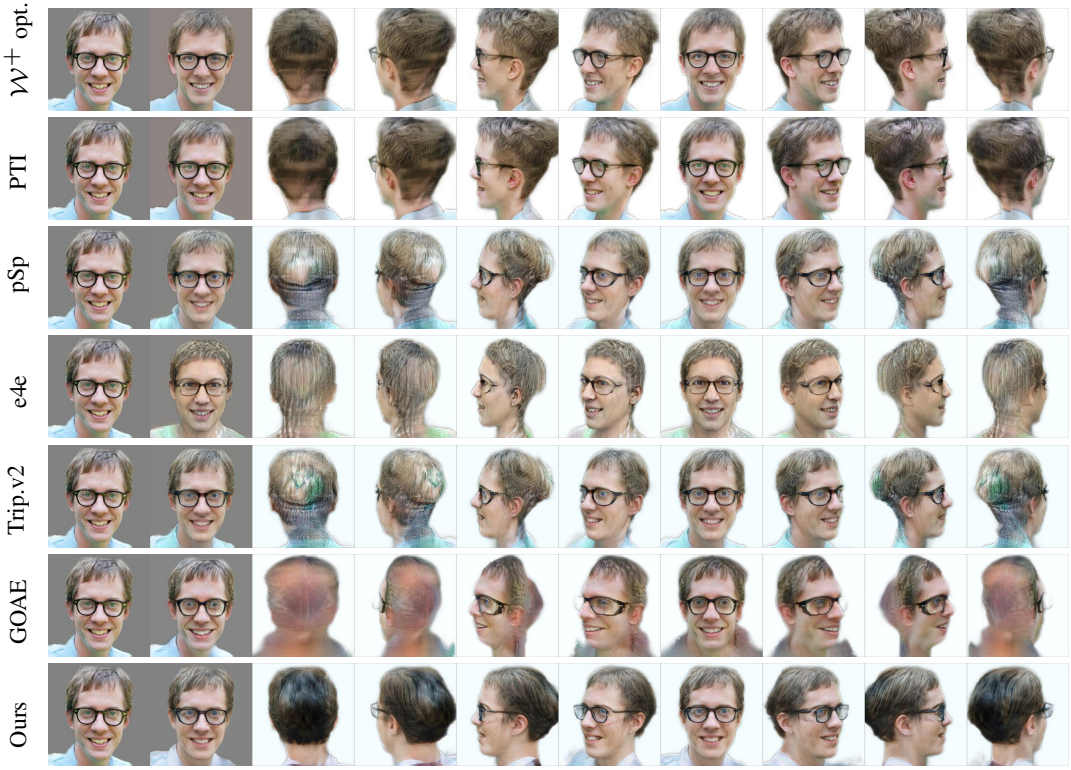

Figure 18: Comparisons of ours and competing methods.

https://doi.org/10.52202/079017-2772

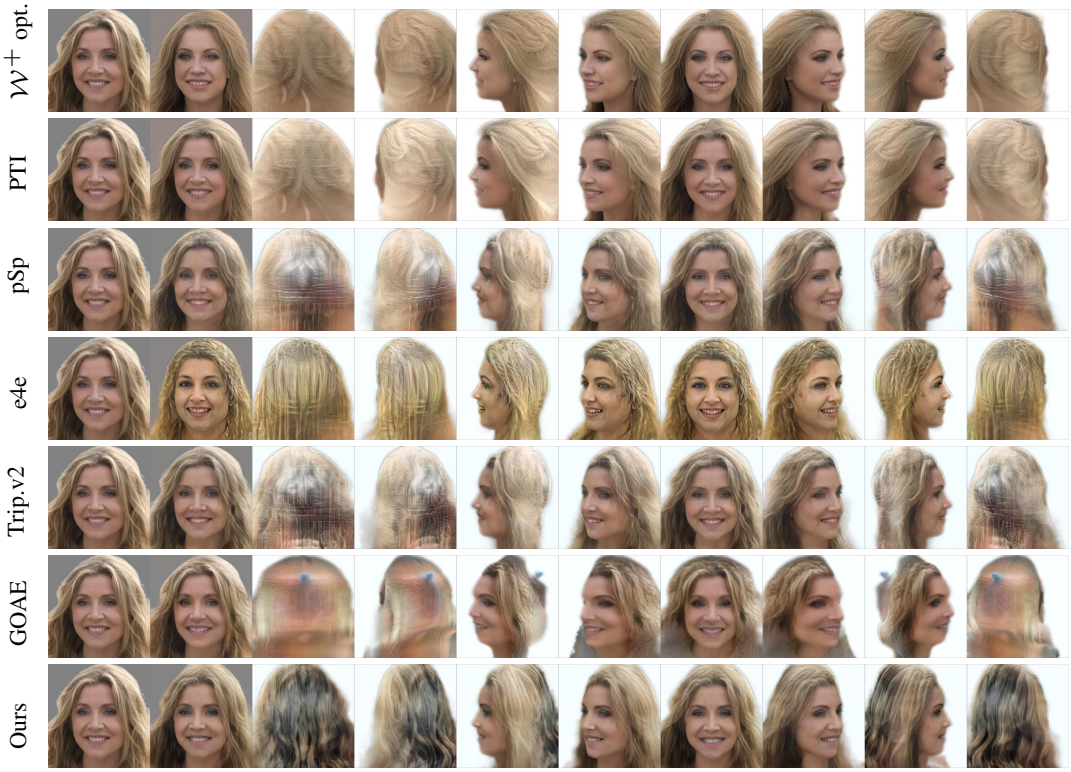

Figure 19: Comparisons of ours and competing methods.

Figure 20: Comparisons of ours and competing methods.

https://doi.org/10.52202/079017-2772

Figure 21: Qualitative results of ablation on occlusion-aware discriminator $\mathcal{D}$.

Figure 22: Visual results of Encoder 1, Encoder 2, and Dual encoders for the given input images in the first column.

Figure 23: Visual results of Encoder 1, Encoder 2, and Dual encoders for the given input images in the first column.

https://doi.org/10.52202/079017-2772

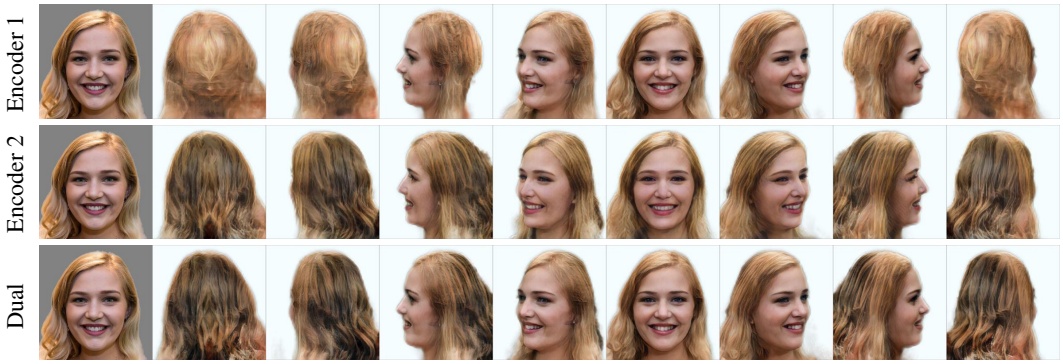

Figure 24: Visual results of Encoder 1, Encoder 2, and Dual encoders for the given input images in the first column.

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: Our claims include utilizing dual encoders trained to specialize in visible and occluded regions, along with a stitching pipeline that seamlessly combines the most accurate predictions from both encoders. To further enhance the reconstruction of occluded regions, we propose an occlusion-aware discriminator, enabling the image encoder to generate realistic features for these challenging areas. Extensive experiments validate the effectiveness of our approach, demonstrating superior performance across multiple benchmarks and scenarios.

   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: Limitations are given in Sec. 6.

   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.

https://doi.org/10.52202/079017-2772

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]

   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: The training and evaluation schemes are detailed in Sections 4.1 and 4.3 and Appendix A, which provides reproducability.

   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: [No]

   Justification: The paper does not provide open access to the code as of now; however, the datasets are public and can be obtained from the pointed references. The code is planned to be released through the project page provided in Abstract.

   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: Training objectives are given in Eqs. (2) and (4), the training recipe in Section 4.1, and the evaluation metrics in Section 4.3. Further details about the experimental setup are provided in Appendix A.

   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: We do not provide error bars due to the computational budget it would require.

   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.

https://doi.org/10.52202/079017-2772

- 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: Time of execution of methods are provided in Table 1, as well as the computing resources in Section 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 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: Broader impacts are provided in Section 6.

    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 Section 6.

    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 mentioned the original owners of previous work via references, and complied with the licenses.

    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.

https://doi.org/10.52202/079017-2772

- 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 share new assets as of now.

    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: The 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.

https://doi.org/10.52202/079017-2772