3D-aware Blending with Generative NeRFs

Hyunsu Kim [1] Gayoung Lee [1] Yunjey Choi [1] Jin-Hwa Kim [1,2] Jun-Yan Zhu[3]
[1]NAVER AI Lab [2]SNU AIIS [3]CMU

Abstract

Image blending aims to combine multiple images seamlessly. It remains challenging for existing 2D-based methods, especially when input images are misaligned due to differences in 3D camera poses and object shapes. To tackle these issues, we propose a 3D-aware blending method using generative Neural Radiance Fields (NeRF), including two key components: 3D-aware alignment and 3D-aware blending. For 3D-aware alignment, we first estimate the camera pose of the reference image with respect to generative NeRFs and then perform pose alignment for objects. To further leverage 3D information of the generative NeRF, we propose 3D-aware blending that utilizes volume density and blends on the NeRF's latent space, rather than raw pixel space. Collectively, our method outperforms existing 2D baselines, as validated by extensive quantitative and qualitative evaluations with FFHQ and AFHQ-Cat.

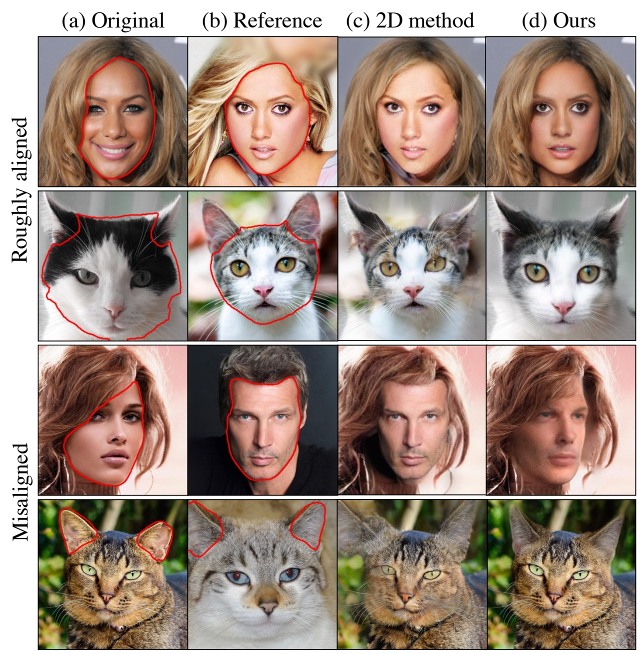


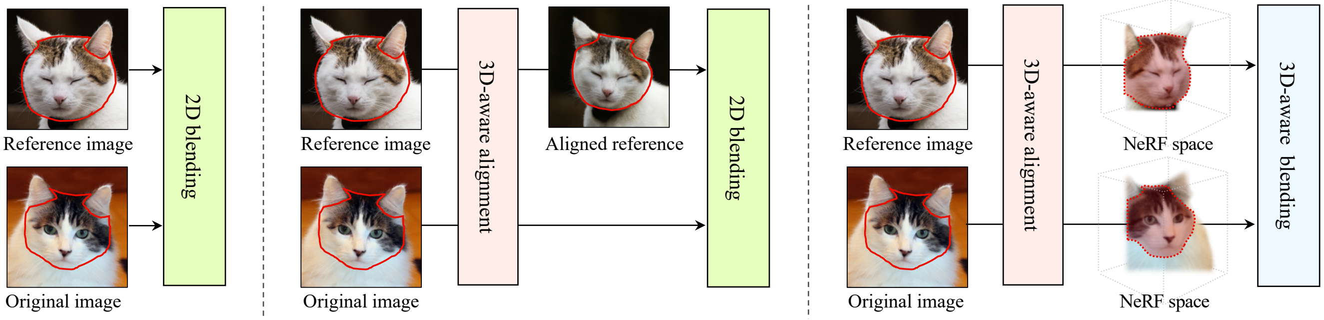
Figure 1: Image blending is challenging for unaligned original and reference images. Existing 2D-based methods [42] struggle to synthesize realistic results due to the 3D object pose differences between foreground and background. In contrast, we propose a 3D-aware blending method that aligns and composes unaligned images without manual effort.

1. Introduction

Image blending aims at combining elements from multiple images naturally, enabling a wide range of applications in content creation, and virtual and augmented realities [95, 96]. However, blending images seamlessly requires delicate adjustment of color, texture, and shape, often requiring users' expertise and tedious manual processes. To reduce human efforts, researchers have proposed various automatic image blending algorithms, including classic methods [62, 49, 7, 76] and deep neural networks [93, 79, 54].

Despite significant progress, blending two unaligned images remains a challenge. Current 2D-based methods often assume that object shapes and camera poses have been accurately aligned. As shown in Figure 1c, even slight misalignment can produce unnatural results, as it is obvious to human eyes that foreground and background objects were captured using different cameras. Several methods [34, 52, 12, 66, 86] warp an image via 2D affine transformation. However, these approaches do not account for 3D geometric differences, such as out-of-plane rotation and 3D shape differences. 3D alignment is much more difficult for users and algorithms, as it requires inferring the 3D structure from a single view. Additionally, even though previous methods get aligned images, they blend images in 2D space. Blending images using only 2D signals, such as pixel values (RGB) or 2D feature maps, doesn't account for the 3D structure of objects.

To address the above issues, we propose a 3D-aware image blending method based on generative Neural Radiance Fields (NeRFs) [9, 33, 10, 59, 67, 91]. Generative NeRFs learn to synthesize images in 3D using only collections of single-view images. Our method projects the input images to the latent space of generative NeRFs and performs 3D-aware alignment by novel view synthesis. We then perform blending on NeRFs' latent space. Concretely, we formulate an optimization problem in which a latent code is optimized to synthesize an image and volume density of the foreground



(a) 2D blending **(b)** 2D blending with our 3D-aware alignment **(c)** Proposed method (Ours)

Figure 2: **Comparison with the existing blending methods.** Red lines denote target blending parts. **(a) 2D blending.** 2D blending methods compose two images without any 3D-aware alignment. **(b) 2D blending with 3D-aware alignment.** To address misalignment, we apply our 3D-aware alignment method to existing 2D blending methods. **(c) Proposed method.** We propose 3D-aware blending after applying our 3D-aware alignment. Note that all methods do not use 3D labels or 3D morphable models.

close to the reference while preserving the background of the original.

Figure 2 shows critical differences between our approach and previous methods. Figure 2a shows a classic 2D blending method composing two 2D images without alignment. We then show the performance of the 2D blending method can be improved using our 3D-aware alignment with generative NeRFs as shown in Figure 2b. To further exploit 3D information, we propose to compose two images in the NeRFs' latent space instead of 2D pixel space. Figure 2c shows our final method.

We demonstrate the effectiveness of our 3D-aware alignment and 3D-aware blending (volume density) on unaligned images. Extensive experiments show that our method outperforms both classic and learning-based methods regarding both photorealism and faithfulness to the input images. Additionally, our method can disentangle color and geometric changes during blending, and create multi-view consistent results. To our knowledge, our method is the first general-purpose 3D-aware image blending method capable of blending a diverse set of unaligned images.

2. Related Work

Image blending aims to compose different visual elements into a single image. Seminal works tackle this problem using various low-level visual cues, such as image gradients [62, 36, 75, 28, 74], frequency bands [7, 6], color and noise transfer [82, 72], and segmentation [49, 65, 1, 51]. Later, researchers developed data-driven systems to compose objects with similar lighting conditions, camera poses, and scene contexts [50, 15, 34].

Recently, various learning-based methods have been proposed, including blending deep features instead of pixels [73, 20, 31] or designing loss functions based on deep features [88, 87]. Generative Adversarial Networks (GAN) have also been used for image blending [79, 92, 21, 42, 8, 95, 68]. For example, In-DomainGAN [92] exploits GAN inversion to achieve seamless blending, and StyleMapGAN [42] blends images in the spatial latent space. Recently, SDEdit [54] proposes a blending method via diffusion models. The above learning-based methods tend to be more robust than pixel-based methods. But given two images with large pose differences, both may struggle to preserve identity or generate unnatural effects.

In specific domains like faces [83, 22, 56, 81] or hair [95, 96, 19, 44], multiple methods can swap and blend unaligned images. However, these methods are limited to faces or hair, and they often need 3D face morphable models [5, 30], or multi-view images [55, 45] to provide 3D information. Our method offers a general-purpose solution that can handle a diverse set of objects without 3D data.

3D-aware generative models. Generative image models learn to synthesize realistic 2D images [32, 70, 26, 77, 14]. However, the original formulations do not account for the 3D nature of our visual world, making 3D manipulation difficult. Recently, several methods have integrated implicit scene representation, volumetric rendering, and GANs into generative NeRFs [67, 10, 57, 25]. Given a sampled viewpoint, an image is rendered via volumetric rendering and fed to a discriminator. For example, EG3D [9] uses an efficient 3D representation called tri-planes, and StyleSDF [58] merges the style-based architecture and the SDF-based volume renderer. Multiple works [9, 58, 91, 33] have developed a two-stage model to generate high-resolution images. With GAN inversion methods [94, 64, 29, 60, 23, 97, 27, 2], we can utilize these 3D-aware generative models to align and blend images and produce multi-view consistent 3D effects.

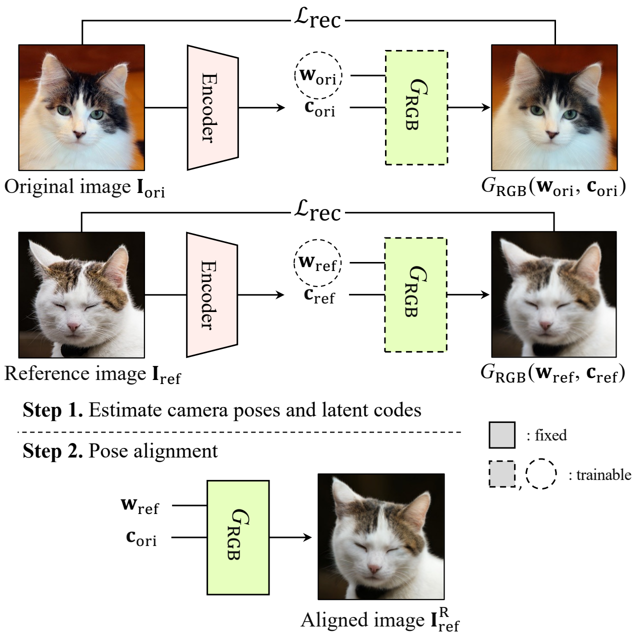


Figure 3: **3D-aware alignment**: we first use a CNN encoder to infer the camera pose of each input image. **Step 1.** Given the camera pose $\mathbf{c}$, we estimate the latent code $\mathbf{w}$ for each input using a reconstruction loss $\mathcal{L}_{\text{rec}}$. **Step 2.** Given the estimated camera pose $\mathbf{c}_{\text{ori}}$ and latent code $\mathbf{w}_{\text{ref}}$, we align the reference image to match the pose of the original image.

3D-aware image editing. Classic 3D-aware image editing methods can create 3D effects given 2D photographs [40, 16, 41]. However, they often require manual efforts to reconstruct the input's geometry and texture. Recently, to reduce manual efforts, researchers have employed generative NeRFs for 3D-aware editing. For example, EditNeRF [53] uses separate latent codes to edit the shape and color of a NeRF object. NeRF-Editing [85] proposes to reflect geometric edits in implicit neural representations. CLIP-NeRF [78] uses a CLIP loss [63] to ensure that the edited result corresponds to the input condition. In SURF-GAN [48], they discover controllable attributes using NeRFs for training a 3D-controllable GAN. Kobayashi et al. [47] enable editing via semantic scene decomposition. While the above works tackle various image editing tasks, we focus on a different task – image blending, which requires both alignment and harmonization. Compared to previous image blending methods, our method addresses blending in a 3D-aware manner.

3. Method

We aim to perform 3D-aware image blending using only 2D images, with target masks from users for both original and reference images. Our method consists of two stages: 3D-aware alignment and 3D-aware blending. Before we blend, we first align the pair of images regarding the pose. In Section 3.1, we describe *pose alignment* for entire objects and *local alignment* for target regions. Then, we apply the 3D-aware blending method in the generative NeRF's latent space in Section 3.2. A variation of our blending method is illustrated in Section 3.3. We combine Poisson blending with our method to achieve near-perfect background preservation. We use EG3D [9] as our backbone, although other 3D-aware generative models, such as StyleSDF [58], can also be applied; see Section E in the supplement.

3.1. 3D-aware alignment

Pose alignment is a requisite process of our blending method, as slight pose misalignment of two images can severely degrade blending quality as shown in Figure 1. To match the reference image I_{ref} to the pose of the original image I_{ori}, we use a generative NeRF G to estimate the camera pose $\mathbf{c}$ and the latent code $\mathbf{w}$ of each image. In *Step 1* in Figure 3, we first train and freeze a CNN encoder (*i.e.*, pose estimation network) to predict the camera poses of input images. During training, we can generate a large number of pairs of camera poses and images using generative NeRF and train the encoder E using a pose reconstruction loss $\mathcal{L}_{\text{pose}}$ as follows:

$$\mathcal{L}_{\text{pose}} = \mathbb{E}_{\mathbf{w},\mathbf{c}} \|\mathbf{c} - E(G_{\text{RGB}}(\mathbf{w}, \mathbf{c}))\|_1, \tag{1}$$

where G_{RGB} is an image rendering function with the generative NeRF G, and $\|\cdot\|_1$ is the L1 distance. The latent code $\mathbf{w}$ and camera pose $\mathbf{c}$ are randomly drawn.

With our trained encoder, we estimate the camera poses $\mathbf{c}_{\text{ori}}$ and $\mathbf{c}_{\text{ref}}$ (defined as Euler angles $\mathbf{c} \in SO3$) of the original and reference images, respectively. Given the estimated camera poses, we project input images I_{ori} and I_{ref} to the latent codes $\mathbf{w}_{\text{ori}}$ and $\mathbf{w}_{\text{ref}}$ using Pivotal Tuning Inversion (PTI) [64]. We optimize the latent code $\mathbf{w}$ using the reconstruction loss $\mathcal{L}_{\text{rec}}$ as follows:

$$\mathcal{L}_{\text{rec}} = \|\mathrm{I} - G_{\text{RGB}}(\mathbf{w}, \mathbf{c})\|_1 + \mathcal{L}_{\text{LPIPS}}(\mathrm{I}, G_{\text{RGB}}(\mathbf{w}, \mathbf{c})), \tag{2}$$

where $\mathcal{L}_{\text{LPIPS}}$ is a learned perceptual image patch similarity (LPIPS) [89] loss. For more accurate inversion, we fine-tune the generator G. Inversion details are described in Section B in the supplement. Finally, as shown in *Step 2* of Figure 3, we can align the reference image as follows:

$$\mathrm{I}^{\text{R}}_{\text{ref}} = G_{\text{RGB}}(\mathbf{w}_{\text{ref}}, \mathbf{c}_{\text{ori}}). \tag{3}$$

While *pose alignment* can align two entire objects, further alignment in editing regions may still be necessary due to variations in scale and translation between object instances. To align target editing regions such as the face, eyes, and ears, we can further employ *local alignment* in the loosely aligned dataset (AFHQv2). The Iterative Closest Point (ICP) algorithm [3, 17] is applied to meshes, which can adjust their scale and translation. For further details, please refer to Section C in the supplement.

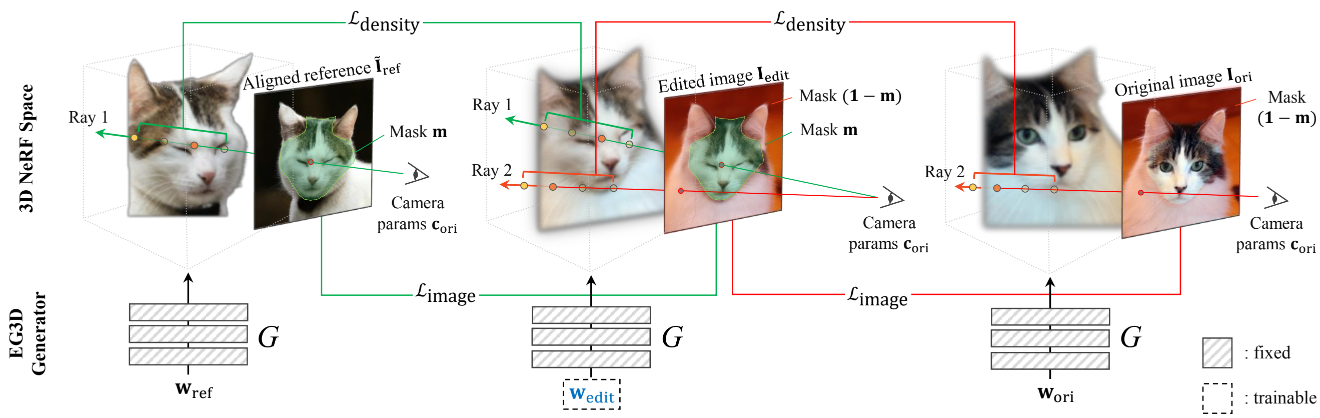


Figure 4: **Our 3D-aware blending pipeline.** We employ density-blending loss ($\mathcal{L}_{\text{density}}$) in the volume density of 3D NeRF space, as well as the image-blending loss ($\mathcal{L}_{\text{image}}$) in 2D image space. Green rays pass through the interior of the mask ($\mathbf{m}$) and red rays pass through the exterior of the mask ($\mathbf{1}-\mathbf{m}$). $\mathcal{L}_{\text{image}}$ and $\mathcal{L}_{\text{density}}$ are used to optimize the latent code $\mathbf{w}_{\text{edit}}$ to generate the well-blended image I_{edit}.

3.2. 3D-aware blending

We aim to find the best latent code $\mathbf{w}_{\text{edit}}$ to synthesize a seamless and natural output. To achieve this goal, we exploit both 2D pixel constraints (RGB value) and 3D geometric constraints (volume density). With the proposed image-blending and density-blending losses, we optimize the latent code $\mathbf{w}_{\text{edit}}$, by matching the foreground with the reference and the background with the original.

Image-blending algorithms are often designed to match the color and details of the original image (*i.e.*, background) while preserving the structure of the reference image (*i.e.*, foreground) [62]. As shown in Figure 4, our image-blending loss matches the color and perceptual similarity of the original image using a combination of L1 and LPIPS [89], while matching the reference image's details using LPIPS loss alone. L1 loss in the reference can lead to overfitting to the pixel space. Let I_{edit} be the rendered image from the latent code $\mathbf{w}_{\text{edit}}$. We define the image-blending loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{image}} &= \|(\mathbf{1}-\mathbf{m})\circ \mathrm{I}_{\text{edit}} - (\mathbf{1}-\mathbf{m})\circ \mathrm{I}_{\text{ori}}\|_1 \\ &+ \lambda_1 \mathcal{L}_{\text{LPIPS}}((\mathbf{1}-\mathbf{m})\circ \mathrm{I}_{\text{edit}}, (\mathbf{1}-\mathbf{m})\circ \mathrm{I}_{\text{ori}}) \\ &+ \lambda_2 \mathcal{L}_{\text{LPIPS}}(\mathbf{m}\circ \mathrm{I}_{\text{edit}}, \mathbf{m}\circ \mathrm{I}_{\text{ref}}), \end{aligned} \tag{4}$$

where $\circ$ denotes element-wise multiplication. Here, λ_1 and λ_2 balance each loss term.

Density-blending is our key component in 3D-aware image blending. If we use only image-blending loss, the blending result easily falls blurry and may not reflect the reference object correctly. Especially, a highly structured object such as hair is hard to be blended in the 3D NeRF space without volume density, as shown in Figure 8. By representing each image as a NeRF instance, we can calculate the density $\boldsymbol{\sigma}$ of a given 3D location $\boldsymbol{x} \in \mathbb{R}^3$. Let $\mathcal{R}_{\text{ref}}$ and $\mathcal{R}_{\text{ori}}$ be the set of rays $\boldsymbol{r}$ passing through the interior and exterior of the target mask $\mathbf{m}$, respectively. For the 3D sample points along the rays $\mathcal{R}_{\text{ref}}$, we aim to match the density field between the reference and our output result, as shown as the sample points in a green ray in Figure 4. For 3D sample points in $\mathcal{R}_{\text{ori}}$, we also match the density field between the original and the result, as shown as the sample points in a red ray in Figure 4. Let $G_\sigma(\mathbf{w};\boldsymbol{x})$ be the density of a given 3D point $\boldsymbol{x}$ with a given latent code $\mathbf{w}$. Our density-blending loss can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{density}} &= \sum_{\boldsymbol{r}\in\mathcal{R}_{\text{ref}}} \sum_{\boldsymbol{x}\in\boldsymbol{r}} \|G_\sigma(\mathbf{w}_{\text{edit}};\boldsymbol{x}) - G_\sigma(\mathbf{w}_{\text{ref}};\boldsymbol{x})\|_1 \\ &+ \sum_{\boldsymbol{r}\in\mathcal{R}_{\text{ori}}} \sum_{\boldsymbol{x}\in\boldsymbol{r}} \|G_\sigma(\mathbf{w}_{\text{edit}};\boldsymbol{x}) - G_\sigma(\mathbf{w}_{\text{ori}};\boldsymbol{x})\|_1. \end{aligned} \tag{5}$$

Our final objective function includes both image-blending loss and density-blending loss:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{image}} + \mathcal{L}_{\text{density}}, \tag{6}$$

where λ is the hyperparameter that controls the contribution of the image-blending loss. If our user wants to blend the shape together without reflecting the color of reference, λ_2 in Eqn. 4 is set to 0. Otherwise, we can set λ_2 to a positive number to reflect the reference image's color and geometry as shown in Figure 9.

3.3. Combining with Poisson blending

While our method produces high-quality blending results, incorporating Poisson blending [62] further improves the preservation of the original image details. Figure 5 shows the effect of Poisson blending with our method. We perform

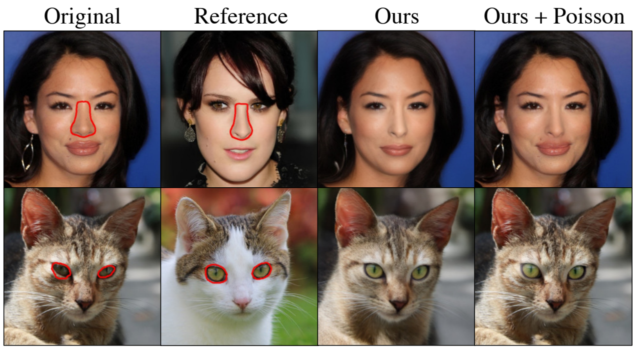


Figure 5: **Ours with Poisson blending** [62]. Ours shows satisfying blending results but a lack of preservation in details. In the first row, the earring is missing in our method. The high-frequency details such as hair and fur are less pronounced in our method alone compared to when it is combined with Poisson blending.

Poisson blending between the original image and the blended image generated by our 3D-aware blending method. Our blending method is modified in two ways. 1) In the initial blending stage, we only preserve the area around the border of the mask instead of all parts of the original image, as we can directly use the original image in the Poisson blending stage. We can reduce the number of iterations from 200 to 100, as improved faithfulness scores are easily achieved; see mL_2 and LPIPS$_m$ in Tables 1 and 2. 2) Instead of using the latent code of the original image $\mathbf{w}_{\text{ori}}$ as the initial value of $\mathbf{w}_{\text{edit}}$, we use the latent code of the reference image $\mathbf{w}_{\text{ref}}$. This allows us to instantly reflect the identity of the reference image and only optimize $\mathbf{w}_{\text{edit}}$ to reconstruct a small region near the mask boundary of the original image. Note that this is an optional choice, as our method *without* Poisson blending has already outperformed all the baselines, as shown in Tables 3 and 4.

4. Experiments

In this section, we show the advantage of our full method over several existing methods and ablated baselines. In Section 4.1, we describe our experimental settings, including baselines, datasets, and evaluation metrics. In Section 4.2, we show both quantitative and qualitative comparisons. In addition to the automatic quantitative metrics, our user study shows that our method is preferred over baselines regarding photorealism. In Section 4.3, we analyze the effectiveness of each module via ablation studies. Lastly, Section 4.4 shows useful by-products of our method, such as generating multi-view images and controlling the color and geometry disentanglement. Please see the supplement for experimental details, video results on the webpage, additional results, *etc*.

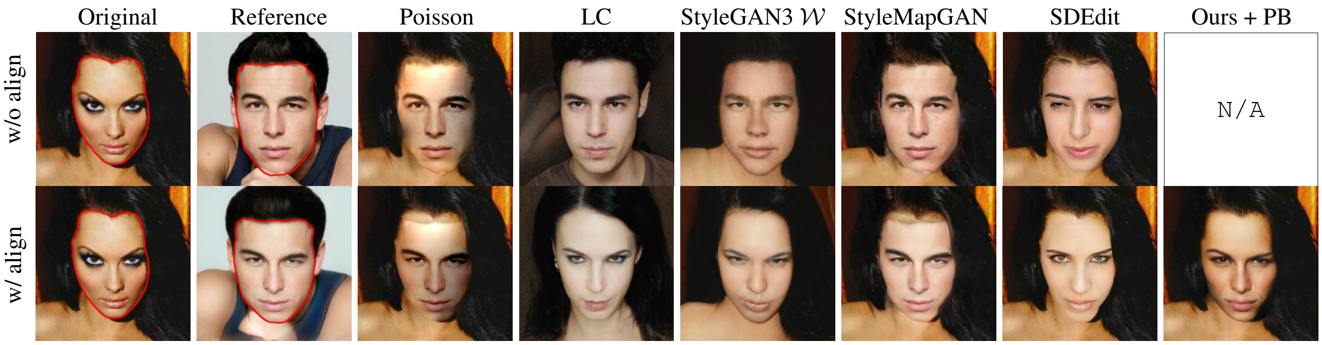
4.1. Experimental setup

Baselines. We compare our method with various image blending methods using only 2D input images. For classic methods, we run Poisson blending [62], a widely-used gradient-domain editing method. We also compare with several recent learning-based methods [8, 42, 38, 54]. Latent Composition [8] utilizes the compositionality in GANs by finding the latent code of the roughly collaged inputs on the manifold of the generator. StyleMapGAN [42] proposes the spatial latent space for GANs to enable local parts blending by mixing the spatial latent codes. Recently, Karras *et al*. [38] proposed StyleGAN3, which provides rotation equivariance. Therefore, we additionally show their blending results by finding the latent code of the composited inputs on the StyleGAN3-R manifold. Both $\mathcal{W}$ and $\mathcal{W}+$ of StyleGAN3-R latent spaces are tested. SDEdit [54] is a diffusion-based blending method that produces a natural-looking result by denoising the corrupted image of a composite image.

Datasets. We use FFHQ [39] and AFHQv2-Cat datasets [18] for model training. We use pose alignment for both datasets and apply further local alignment to the loosely aligned dataset (AFHQ).

To test blending performance, we use CelebA-HQ [37] for the FFHQ-trained models and AFHQv2-Cat test sets for the AFHQ-trained models. We randomly select 250 pairs of images from each dataset for an original and reference image. We also create a target mask for each pair to automatically simulate a user input using pretrained semantic segmentation networks [84, 90, 13]. We blend 5 and 3 semantic parts in each pair of images for CelebA-HQ and AFHQ, respectively. The total number of blended images in each method is 1,250 (CelebA-HQ) and 750 (AFHQv2-Cat). We also include results on ShapeNet-Car dataset [11] to show that our method works well for non-facial data.

Evaluation metrics. For evaluation metrics, we use masked L_2, masked LPIPS [89] and Kernel Inception Score (KID) [4]. Masked L_2 (mL_2) is the L_2 distance between the original image and the blended image on the exterior of the mask, measuring the preservation of non-target areas of the original image. Unlike background regions, a pixel-wise loss is too strict for the target area changed during blending. We measure the perceptual similarity metric (LPIPS) [89] for the blended regions, which are called masked LPIPS (LPIPS$_m$) used in previous methods [35, 54]. Kernel Inception Score (KID) [4] is widely used to quantify the realism of the generated images regarding the real data distribution. We compute KID between blended images and the training dataset using the `clean-fid` library [61].



Method	w/o align (baseline only)			w/ 3D-aware align		
	KID ↓	LPIPS$_m$ ↓	mL_2 ↓	KID ↓	LPIPS$_m$ ↓	mL_2 ↓
Poisson Blending [62]	<u>0.006</u>	0.4203	0.0069	<u>0.005</u>	0.2355	0.0051
Latent Composition [8]	0.012	0.4735	0.0388	0.012	0.4487	0.0321
StyleGAN3 $\mathcal{W}$ [38]	0.016	0.4379	0.0353	0.017	0.3921	0.0307
StyleGAN3 $\mathcal{W}+$ [38]	0.025	0.4634	0.0462	0.023	0.4086	0.0391
StyleMapGAN (32 × 32) [42]	0.007	0.3792	0.0118	0.006	<u>0.1989</u>	0.0045
SDEdit [54]	0.011	0.3857	0.0076	0.008	0.3427	**0.0003**
Ours	0.013	<u>0.2046</u>	<u>0.0050</u>	0.013	0.2046	0.0050
Ours + Poisson Blending	**0.002**	**0.1883**	**0.0007**	**0.002**	**0.1883**	<u>0.0007</u>

Table 1: **Comparison with baselines in the CelebA-HQ test set**. The first and second rows of the *figure* show the blending results without and with our 3D-aware alignment, respectively. Metric scores on the left side of the *table* show the results without alignment. We apply our 3D-aware alignment to the baselines on the right side of the table. Lower scores denote better performance in all metrics. The best and second-best scores are bold and underlined. Our method outperforms baselines in all metrics. LC and PB stand for Latent Composition [8] and Poisson Blending [62], respectively. Note that our method always operates 3D-aware alignment, as it is an integral part of our algorithm.

User study. To further examine the effectiveness of our 3D-aware blending method, we conduct a user study for photorealism. Our target goal is to edit the original image, so we exclude baselines that show highly flawed preservation of the original image. Human evaluates pairwise comparison of blended images between our method and one of the baselines. The user selects more real-looking images. We collect 5,000 comparison results via Amazon Mechanical Turk (MTurk).

4.2. Comparison with baselines

Here we compare our method with baselines in two variations. In the *w/o align* setting, we do not apply our 3D-aware alignment to baselines. In the *w/ align* setting, we align the reference image with our 3D-aware alignment. This experiment demonstrates the effectiveness of our proposed method. 1) Our alignment method consistently improves all baselines in all evaluation metrics: KID, LPIPS$_m$, and masked L_2. 2) Our 3D-aware blending method outperforms all baselines, including those that use our alignment method. We also report the combination of our method and Poisson blending to achieve better background preservation, as the perfect inversion is still hard to be achieved in GAN-based methods.

Table 1 shows comparison results in CelebA-HQ. The left side of the table includes all the baselines without our 3D-aware alignment. All metrics are worse than the right side of the table (w/ alignment). This result reveals that alignment between the original and reference image affects overall editing performance. Table 2 shows comparison results in AFHQv2-Cat. It shows the same tendency as Table 1.

Our method performs well regarding all metrics. Combined with Poisson blending, our method outperforms all baselines. Poisson blending and StyleMapGAN (16 × 16, 32 × 32) show great faithfulness to the input images but suffer from artifacts. Latent Composition, StyleMapGAN (8 × 8), and StyleGAN3 $\mathcal{W}$ produce realistic results but far from the input images. The identities of the original and reference images have changed, which is reflected by a worse LPIPS$_m$ and mL_2. SDEdit fails to reflect the reference image and shows worse LPIPS$_m$. StyleGAN3 $\mathcal{W}+$ often shows entirely collapsed images. Our method preserves the identity of the original image and reflects the reference image well while producing realistic outputs.

User study. We note that KID has a high correlation with background preservation. Unfortunately, it fails to capture the boundary artifacts and foreground image quality, espe-

Method	w/o align (baseline only) KID ↓	LPIPS$_m$ ↓	mL_2 ↓	w/ 3D-aware align KID ↓	LPIPS$_m$ ↓	mL_2 ↓
Poisson Blending [62]	**0.002**	0.4956	0.0024	**0.002**	0.2656	**0.0004**
StyleGAN3 $\mathcal{W}$ [38]	0.006	0.4588	0.0316	0.006	0.3802	0.0268
StyleGAN3 $\mathcal{W}+$ [38]	0.014	0.4941	0.0298	0.013	0.3903	0.0236
StyleMapGAN (8×8) [42]	0.013	0.4840	0.0574	0.013	0.3221	0.0526
StyleMapGAN (16×16) [42]	0.006	0.4746	0.0225	0.004	0.2707	0.0160
Ours	0.005	0.2739	0.0073	0.005	0.2739	0.0073
Ours + Poisson Blending	**0.002**	**0.2229**	**0.0013**	**0.002**	**0.2229**	0.0013

Table 2: **Comparison with baselines in the AFHQv2-Cat test set**. Formats of the figure and table are the same as Table 1.

Method	**Ours** w/o	w/ align	**Ours + Poisson Blending** w/o	w/ align
Poisson [62]	79.9%	59.9%	80.9% (+1.0)	67.5% (+7.6)
StyleMap [42]	72.3%	62.0%	75.4% (+3.1)	66.3% (+4.3)
SDEdit [54]	61.0%	55.7%	61.1% (+0.1)	50.2% (-5.5)

Table 3: **User study in CelebA-HQ** regarding the photorealism of the blended image. The percentage denotes how often MTurk workers prefer our method to each baseline in pairwise comparison. Values larger than 50% mean ours outperforms the baseline. Our method, both with and without Poisson blending, outperforms all baselines even if we improve the baselines using our 3D-aware alignment. Incorporating Poisson blending further enhances the realism score of our method as shown in green numbers.

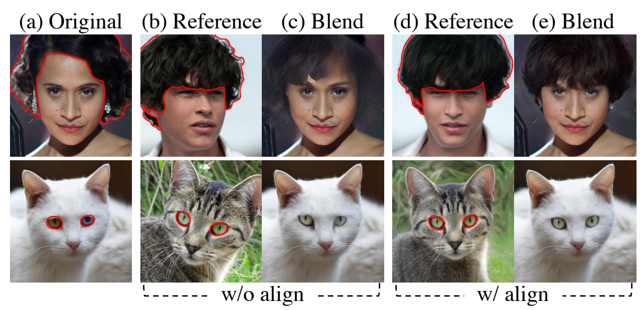


Figure 6: **The effect of our 3D-aware alignment**. Aligned reference images (d) have the same pose as the original images (a). With our alignment, blending results (e) look more realistic and reflect the reference well than those without alignment (c).

Method	**Ours** w/o	w/ align	**Ours + Poisson Blending** w/o	w/ align
Poisson [62]	91.2%	76.2%	87.6% (-3.6)	82.8% (+6.6)
StyleMap [42]	91.0%	82.3%	91.6% (+0.6)	83.6% (+1.3)

Table 4: **User study in AFHQv2-Cat** regarding the photorealism of the blended image. All details are the same as in Table 3. Our approach surpasses all baselines, and the incorporation of Poisson blending further improves the realism score.

cially for small foregrounds. To further evaluate the realism score of results, we conduct a human perception study for our method and other baselines, which shows great preservation scores mL_2. As shown in Tables 3 and 4, MTurk participants prefer our method to other baselines regarding the photorealism of the results. Our method, as well as the combination of ours with Poisson blending, outperforms the baselines. SDEdit with our 3D-aware alignment shows a comparable realism score with ours, but it can not reflect the reference well, as reflected in worse LPIPS$_m$ score in Table 1. Similar to Tables 1 and 2, Mturk participants prefer baselines with alignment to their unaligned counterparts.

Figure 7: Ablation study of Poisson blending (PB) in baselines. Despite combining Poisson blending with the baselines, StyleMapGAN still generates artifacts, and other baselines fail to preserve the identity of the reference. Our method with Poisson blending keeps the original image intact while accurately reflecting the reference image.

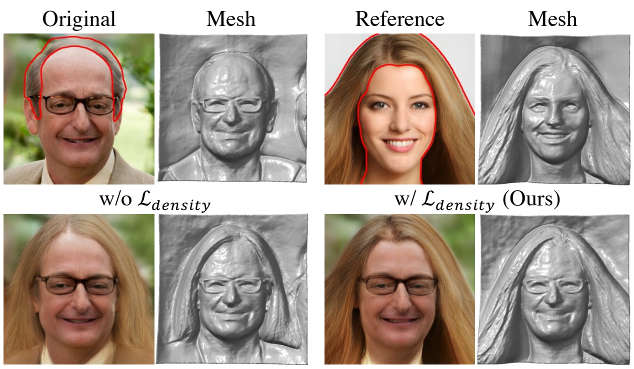


Figure 8: **The effect of our density-blending loss**. Without the loss, 3D information is not considered, resulting in inaccurate blending in 3D space. In the bottom left figure, the hair mesh is not properly reflected without the density-blending loss, resulting in inaccurate blending and missing fine details.

Figure 9: **Color-geometry disentanglement with our model**. We can adjust the reflection of the reference image's color by adjusting the weight λ_2 in the image-blending loss. Without image blending loss on reference, we can focus on object shapes, as shown in the rightmost column.

4.3. Ablation study

3D-aware alignment is an essential part of image blending. As discussed in Section 4.2, our alignment provides consistent improvements in all baseline methods. Moreover, it plays a crucial role in our blending approach. Figure 6 shows the importance of 3D-aware alignment, where the lack of alignment in the reference images result in degraded blending results (Figure 6c). Specifically, the woman's hair appears blurry, and the size of the cat's eyes looks different. Aligned reference images can generate realistic blending results (Figure 6e) in our 3D-aware blending method.

Density-blending loss gives rich 3D signals in the blending procedure. Section 3.2 explains how we can exploit volume density fields in blending. Delicate geometric structures, such as hair, can not be easily blended without awareness of 3D information. Figure 8 shows an ablation study of our density-blending loss. In the bottom left, the hair looks blurry in the blended image, and the mesh of the result shows shorter hair than that in the reference image. In the bottom right, the well-blended image and corresponding mesh show that our density-blending loss contributes to capturing the highly structured object in blending.

Combination with Poisson blending. In Tables 1 and 2, we report the combination of our method and Poisson blending. It shows Poisson blending further enhances the performance of our method in all automatic metrics: KID, mL_2, and LPIPS$_m$. In the realism score of human perception, ours with Poisson blending enhance the score, as shown in green numbers of Tables 3 and 4. However, combining Poisson blending with each baseline does not have meaningful benefits, as shown in Figure 7. Baselines still show artifacts or fail to reflect the identity of the reference.

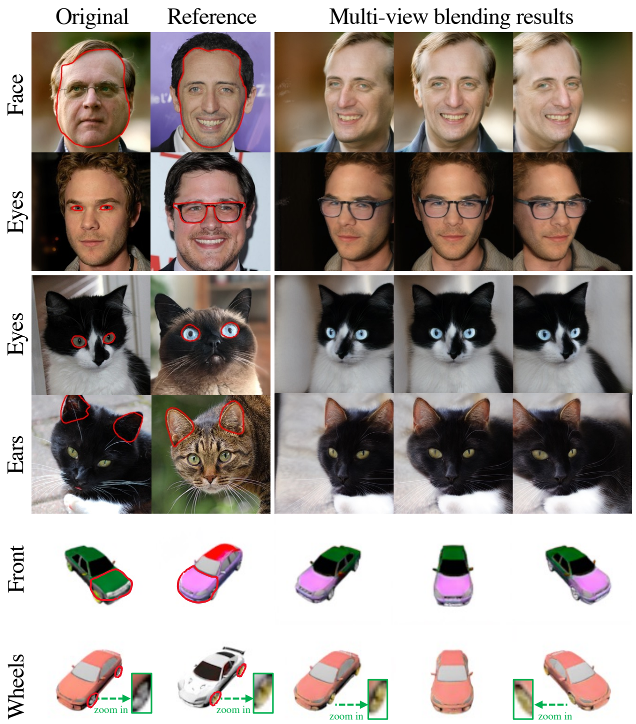


Figure 10: **Multi-view blending results** in various datasets: CelebA-HQ, AFHQv2-Cat, and ShapeNet-Car. Since we optimize the latent code of the generative NeRF, we can synthesis images of the blended object in different poses through the generative NeRF.

4.4. Additional advantages of NeRF-based blending

In addition to increasing blending quality, our 3D-aware method enables additional capacity: color-geometry disentanglement and multi-view consistent blending. As shown in Figure 9, we can control the influence of color in blending. The results with $\mathcal{L}_{\text{image}}$ have a redder color than the results without the loss. If we remove or assign a lower weight to the image-blending loss on reference (λ_2 in Eqn. 4), we can reflect the geometry of the reference object more than the color. In contrast, we can reflect colors better if we give a larger weight to λ_2. Note that we always use the image-blending loss on the original image to preserve it better.

A key component of generative NeRFs is multi-view consistent generation. After applying the blending procedure described in Section 3.2, we have an optimized latent code $\mathbf{w}_{\text{edit}}$. Generative NeRF can synthesize a novel view blended image using $\mathbf{w}_{\text{edit}}$ and a target camera pose. Figure 10 shows the multi-view consistent blending results in CelebA-HQ, AFHQv2-Cat, and ShapeNet-Car [11]. In Section I in the supplement and the attached website, we provide more multi-view blending results and videos for EG3D and StyleSDF [58].

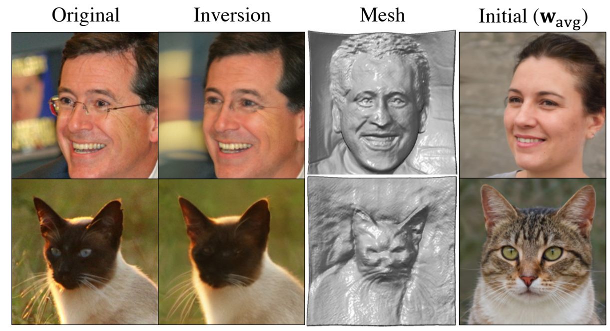


Figure 11: Failure cases of inversion. If an input image has a large variance in scale relative to the mean face or the estimated pose from the encoder is not valid, inversion sometimes fails. The first row shows a failure to reconstruct eyeglasses, and the second row shows a crushed face of a cat in the reconstructed image and mesh.

5. Discussion and Limitations

Our method exploits the capability of NeRFs to align and blend images in a 3D-aware manner only with a collection of 2D images. Our 3D-aware alignment boosts the quality of existing 2D baselines. 3D-aware blending exceeds improved 2D baselines with our alignment method and shows additional advantages such as color-geometry disentanglement and multi-view consistent blending. We hope our approach paves the road to 3D-aware blending. Recently, 3DGP [69] presents a 3D-aware GAN, handling non-alignable scenes captured from arbitrary camera poses in real-world environments. Since our approach relies solely on a pre-trained generator, it can be readily extended to blend unaligned multi-category datasets such as ImageNet [24].

Despite improvements over existing blending baselines, our method depends on GAN inversion, which is a bottleneck of the overall performance regarding quality and speed. Figure 11 shows the inversion process can sometimes fail to accurately reconstruct the input image. We cannot obtain an acceptable inversion result if an input image is far from the average face generated from the mean latent code $\mathbf{w}_{\text{avg}}$. We also note the camera pose inferred by our encoder should not be overly inaccurate. Currently, the problem is being addressed by combining our method with Poisson blending. However, more effective solutions may be available with recent advances in 3D GAN inversion techniques [80, 46]. In the future, to enable real-time editing, we could explore training an encoder [27, 42] to blend images using our proposed loss functions.

Acknowledgments. We would like to thank Seohui Jeong, Che-Sang Park, Eric R. Chan, Junho Kim, Jung-Woo Ha, Youngjung Uh, and other NAVER AI Lab researchers for their helpful comments and sharing of materials. All experiments were conducted on NAVER Smart Machine Learning (NSML) platform [43, 71].

References

[1] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *ACM SIGGRAPH*. 2004. 2

[2] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020. 2

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, 1992. 3

[4] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations (ICLR)*, 2018. 5

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999. 2

[6] Matthew Brown, David G Lowe, et al. Recognising panoramas. In *IEEE International Conference on Computer Vision (ICCV)*, 2003. 2

[7] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*. 1987. 1, 2

[8] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 5, 6

[9] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3

[10] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5, 9

[12] Bor-Chun Chen and Andrew Kae. Toward realistic image compositing with adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[14] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning (ICML)*, 2020. 2

[15] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *ACM Transactions on graphics (TOG)*, 2009. 2

[16] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on graphics (TOG)*, 2013. 3

[17] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 1992. 3

[18] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[19] Chaeyeon Chung, Taewoo Kim, Hyelin Nam, Seunghwan Choi, Gyojung Gu, Sunghyun Park, and Jaegul Choo. Hairfit: pose-invariant hairstyle transfer via flow-based hair alignment and semantic-region-aware inpainting. In *The British Machine Vision Conference (BMVC)*, 2021. 2

[20] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[21] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[22] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. In *ACM SIGGRAPH Asia*, 2011. 2

[23] Giannis Daras, Wen-Sheng Chu, Abhishek Kumar, Dmitry Lagun, and Alexandros G Dimakis. Solving inverse problems with nerfgans. *arXiv preprint arXiv:2112.09061*, 2021. 2

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 9

[25] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[26] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2

[27] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 9

[28] Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. Coordinates for instant image cloning. *ACM Transactions on graphics (TOG)*, 2009. 2

[29] Qianli Feng, Viraj Shah, Raghudeep Gadde, Pietro Perona, and Aleix Martinez. Near perfect gan inversion. *arXiv preprint arXiv:2202.11833*, 2022. 2

[30] Claudio Ferrari, Stefano Berretti, Pietro Pala, and Alberto Del Bimbo. A sparse and locally coherent morphable face model for dense semantic correspondence across heterogeneous 3d faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 2

[31] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. Tilegan: synthesis of large-scale non-homogeneous textures. *ACM Transactions on graphics (TOG)*, 2019. 2

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 2

[33] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[34] James Hays and Alexei A Efros. Scene completion using millions of photographs. *ACM Transactions on graphics (TOG)*, 2007. 1, 2

[35] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 17–34. Springer, 2020. 5

[36] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. In *ACM Transactions on graphics (TOG)*, 2006. 2

[37] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 5

[38] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 5, 6, 7

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[40] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on graphics (TOG)*, 2011. 3

[41] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on graphics (TOG)*, 2014. 3

[42] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 5, 6, 7, 9

[43] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 9

[44] Taewoo Kim, Chaeyeon Chung, Yoonseo Kim, Sunghyun Park, Kangyeol Kim, and Jaegul Choo. Style your hair: Latent optimization for pose-invariant hairstyle transfer via local-style-aware hair alignment. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[45] Taewoo Kim, Chaeyeon Chung, Sunghyun Park, Gyojung Gu, Keonmin Nam, Wonzo Choe, Jaesung Lee, and Jaegul Choo. K-hairstyle: A large-scale korean hairstyle dataset for virtual hair editing and hairstyle classification. In *IEEE International Conference on Image Processing (ICIP)*, 2021. 2

[46] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. *WACV*, 2023. 9

[47] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 3

[48] Jeong-gi Kwak, Yuanming Li, Dongsik Yoon, Donghyeon Kim, David Han, and Hanseok Ko. Injecting 3d perception of controllable nerf-gan into stylegan for editable portrait image synthesis. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[49] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on graphics (TOG)*, 2003. 1, 2

[50] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM Transactions on graphics (TOG)*, 2007. 2

[51] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2007. 2

[52] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[53] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[54] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 5, 6, 7

[55] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 2

[56] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 2022. 2

[57] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[58] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 9

[59] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 1

[60] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multi-layer selection for gan inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11399–11409, 2022. 2

[61] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022. 5

[62] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH*, 2003. 1, 2, 4, 5, 6, 7

[63] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 3

[64] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 2022. 2, 3

[65] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM Transactions on graphics (TOG)*, 2004. 2

[66] Othman Sbai, Camille Couprie, and Mathieu Aubry. Surprising image compositions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, 2021. 1

[67] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2

[68] Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[69] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. In *ICLR*, 2023. 9

[70] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[71] Nako Sung, Minkyu Kim, Hyunwoo Jo, Youngil Yang, Jingwoong Kim, Leonard Lausen, Youngkwan Kim, Gayoung Lee, Donghyun Kwak, Jung-Woo Ha, et al. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*, 2017. 9

[72] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010. 2

[73] Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable image synthesis with internal representation collaging. In *arXiv preprint arXiv:1811.10153*, 2018. 2

[74] Richard Szeliski, Matthew Uyttendaele, and Drew Steedly. Fast poisson blending using multi-splines. In *IEEE International Conference on Computational Photography (ICCP)*, 2011. 2

[75] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. In *European Conference on Computer Vision (ECCV)*, 2010. 2

[76] Matthew Uyttendaele, Ashley Eden, and Richard Skeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. 1

[77] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 2

[78] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[79] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Gp-gan: Towards realistic high-resolution image blending. In *ACM international conference on multimedia (ACM-MM)*, 2019. 1, 2

[80] Jiaxin Xie, Hao Ouyang, Jingtan Piao, Chenyang Lei, and Qifeng Chen. High-fidelity 3d gan inversion by pseudo-multi-view optimization. In *CVPR*, 2023. 9

[81] Yangyang Xu, Bailin Deng, Junle Wang, Yanqing Jing, Jia Pan, and Shengfeng He. High-resolution face swapping via latent semantics disentanglement. In *CVPR*, 2022. 2

[82] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and improving the realism of image composites. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 2

[83] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3d-aware face component transfer. In *ACM SIGGRAPH*, 2011. 2

[84] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[85] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[86] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[87] He Zhang, Jianming Zhang, Federico Perazzi, Zhe Lin, and Vishal M Patel. Deep image compositing. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[88] Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. Deep image blending. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020. 2

[89] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 4, 5

[90] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[91] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *arXiv preprint arXiv:2110.09788*, 2021. 1, 2

[92] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[93] Jun-Yan Zhu, Philipp Krahenbuhl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

[94] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[95] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. In *ACM Transactions on graphics (TOG)*, 2021. 1, 2

[96] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Hairnet: Hairstyle transfer with pose changes. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[97] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*, 2020. 2