SEM2NERF: CONVERTING SINGLE-VIEW SEMANTIC MASKS TO NEURAL RADIANCE FIELDS

Yuedong Chen
Monash University, Australia
yuedong.chen@monash.edu

Qianyi Wu
Monash University, Australia
qianyi.wu@monash.edu

Chuanxia Zheng
Monash University, Australia
chuanxia001@e.ntu.edu.sg

Tat-Jen Cham
Nanyang Technological University, Singapore
astjcham@ntu.edu.sg

Jianfei Cai
Monash University, Australia
jianfei.cai@monash.edu

ABSTRACT

Image translation and manipulation have gain increasing attention along with the rapid development of deep generative models. Although existing approaches have brought impressive results, they mainly operated in 2D space. In light of recent advances in NeRF-based 3D-aware generative models, we introduce a new task, Semantic-to-NeRF translation, that aims to reconstruct a 3D scene modelled by NeRF, conditioned on one single-view semantic mask as input. To kick-off this novel task, we propose the Sem2NeRF framework. In particular, Sem2NeRF addresses the highly challenging task by encoding the semantic mask into the latent code that controls the 3D scene representation of a pretrained decoder. To further improve the accuracy of the mapping, we integrate a new region-aware learning strategy into the design of both the encoder and the decoder. We verify the efficacy of the proposed Sem2NeRF and demonstrate that it outperforms several strong baselines on two benchmark datasets. Project page: https://donydchen.github.io/sem2nerf

Keywords Semantic to NeRF translation · Conditional generative model · NeRF · 3D-based translation.

1 Introduction

Controllable image generation, translation, and manipulation have seen rapid advances in the last few years along with the emergence of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. Current systems are able to freely change the image appearance through referenced images [Johnson et al., 2016, Zhu et al., 2017a, Isola et al., 2017], modify scene content via semantic masks [Wang et al., 2018a, Park et al., 2019, Ling et al., 2021], and even accurately manipulate various attributes in feature space [Karras et al., 2019, Wu et al., 2021a,b]. Despite impressive performance and wide applicability, these systems are mainly focused on 2D images, without directly considering the 3D nature of the world and the objects within.

Concurrently, significant progress has been made for 3D generation by using deep generative networks [Goodfellow et al., 2014, Kingma and Ba, 2014]. Methods were developed for different 3D shape representations, including voxels [Wu et al., 2016], point clouds [Luo and Hu, 2021], and meshes [Goel et al., 2020]. More recently, Neural Radiance Fields (NeRF) [Mildenhall et al., 2020] has been a new paradigm for 3D representation, providing accurate 3D shape and view-dependent appearance simultaneously. Based on this new representation, seminal 3D generation approaches [Schwarz et al., 2020, Niemeyer and Geiger, 2021, Chan et al., 2021a, Gu et al., 2021, Chan et al., 2021b] have been proposed that aim to generate photorealistic images from a given distribution in a 3D-aware and view-consistent manner. However, these techniques are primarily developed purely for high-quality 3D generation, leaving controllable 3D manipulation and editing unsolved.

It would be a dramatic enhancement if we can *freely manipulate and edit an object's content and appearance in 3D space, while only leveraging easily obtained 2D input information*. In this paper, we take an initial step toward this

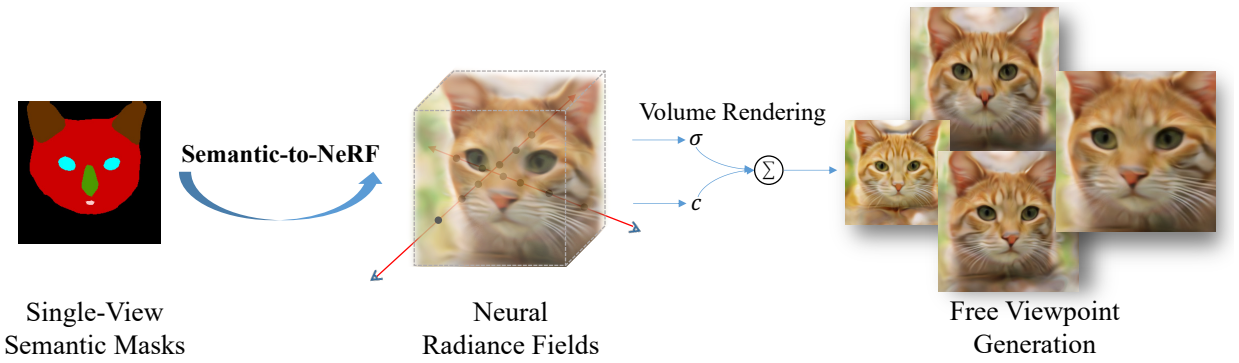

Figure 1: Illustration of the Semantic-to-NeRF translation task, which aims to achieve free-viewpoint image generation by taking only a single-view semantic mask as input

grand goal by introducing a new task, termed **Semantic-to-NeRF translation**, analogous to a 2D Semantic-to-Image translation task but operating on 3D space. Specifically, as illustrated in Figure 1, Semantic-to-NeRF translation can take as input a single-view 2D semantic mask, yet output a NeRF-based 3D representation that can be used to render photorealistic images in a 3D-aware view-consistent manner. More importantly, it allows free editing of the object's content and appearance in 3D space, by modifying the content only via a single-view 2D semantic mask.

However, generating 3D structure from a single 2D image is already an ill-posed problem, and it will be even more so from a single 2D semantic mask. There are also two other major issues in this novel task:

1. *Large information gap between 3D structure and 2D semantics.* A single-view 2D semantic mask neither holds any 3D shape or surface information, nor provides much guidance for plausible appearances, making it tough to generate a neural radiance field with comprehensive details.
2. *Imbalanced semantic distribution.* Since semantic classes tend to be area-imbalanced within an image, *e.g.* eyes occupy less than 1% of a face while hair can take up larger than 40%, existing CNN-based networks may over-attend to larger semantic regions, while discounting smaller semantic regions that may be perceptually more salient. This will result in poor controllable editing in 3D space when we alter small semantic regions.

To mitigate these issues, we propose a novel framework, called **Sem2NeRF**, that builds on NeRF [Mildenhall et al., 2020] for 3D representation, by augmenting it with a *semantic translation branch* that conditionally generates high-quality 3D-consistent images. In particular, the proposed framework is based on an encoder-decoder architecture that converts a singe-view 2D semantic mask to an embedded code, and then transfers it to a NeRF representation for rendering photorealistic 3D-consistent images.

Our broad idea here is that instead of directly learning to predict 3D structure from degenerate single-view 2D semantic masks, *the network can alternatively learn the 3D shape and appearance representation from large numbers of unstructured 2D RGB images*. This has achieved significant advances in NeRF-based generator [Schwarz et al., 2020, Niemeyer and Geiger, 2021, Chan et al., 2021a, Gu et al., 2021, Chan et al., 2021b], which transforms a random vector to a NeRF representation. In short, our scenario is thus: we have a well-trained 3D generator, but we aim to further control the generated content and appearance easily. The main idea is then to *learn a good mapping network* (like current methods for 2D GAN inversion [Richardson et al., 2021, Song et al., 2021]) *that can encode the semantic mask into the somewhat smaller latent space domain for 3D controllable translation and manipulation.* As for the second issue, we intriguingly discover that a *region-aware learning strategy* is of vital importance. We therefore aim to tame an encoder that is sensitive to image patches, and adopt a region-based sampling pattern for the decoder. Furthermore, augmenting the input semantic masks with extracted contours and distance field representations [Chen and Hays, 2018] also considerably helps to highlight the intended semantic changes, making them more easily perceptible.

Following the above analysis, we build our Sem2NeRF framework upon the Swin Transformer encoder [Liu et al., 2021] and the pre-trained π-GAN decoder [Chan et al., 2021a]. To kick off the single-view Semantic-to-NeRF translation task, we pinpoint two suitable yet challenging datasets, including CelebAMask-HQ [Lee et al., 2020] and CatMask, where the latter contains cat faces rendered using π-GAN and labelled with 6-class semantic masks using DatasetGAN [Zhang et al., 2021]. We showcase the superiority of our model over several strong baselines by considering SofGAN [Chen et al., 2022], pix2pixHD [Wang et al., 2018b] with GAN-inversion [Karras et al., 2020], and pSp [Richardson et al., 2021]. Our contributions are three-fold:

- We introduce a novel and challenging task, Semantic-to-NeRF translation, which converts a single-view 2D semantic mask to a 3D scene modelled by neural radiance fields.

- With the insight of needing a region-aware learning strategy, we propose a novel framework, Sem2NeRF, which is capable of achieving 3D-consistent free viewpoint image generation, semantic editing and multi-model synthesis with a given single-view semantic mask.

- We validate our insight regarding our region-aware learning strategy and the efficacy of Sem2NeRF via extensive ablation studies, and demonstrate that Sem2NeRF outperforms strong baselines on two challenging datasets.

2 Related Work

NeRF and Generative NeRF. Starting as an approach focused on modelling a single static scene, NeRF [Mildenhall et al., 2020] had seen rapid development in different aspects. Several approaches managed to reduce the training [Sun et al., 2021] and inference time [Liu et al., 2020, Lombardi et al., 2021], while others improved visual quality [Barron et al., 2021]. Besides, it had also been extended in other ways, *e.g.*, dynamic scenes [Pumarola et al., 2021, Park et al., 2021], pose estimation [Yen-Chen et al., 2021], portrait generation [Liu et al., 2022], semantic segmentation [Zhi et al., 2021].

Follow-up works that integrated NeRF with generative models were most relevant to ours. Schwarz *et al*. [Schwarz et al., 2020] proposed to learn a NeRF distribution by conditioning the input point positions with a sampled random vector. Niemeyer *et al*. [Niemeyer and Geiger, 2021] enabled multi-object generation by representing the whole scenes as a composition of different components. To improve visual quality of the generated images, Chan *et al*. [Chan et al., 2021a] proposed π-GAN by replacing the general MLP-based network with a SIREN-based [Sitzmann et al., 2020] network, conditioned with multi-layer latent codes. CIPS-3D [Zhou et al., 2021] aimed to improve image quality by concatenating StyleGAN [Karras et al., 2020] to π-GAN. Similarly, StyleNeRF [Gu et al., 2021] turned to embedding the volume rendering technique into StyleGAN. More recently, EG3D [Chan et al., 2021b] enhanced generative quality with a tri-plane-based design that can leverage a state-of-the-art 2D generative model while operating on a 3D representation.

Our work belongs to the class of generative models, but unlike all existing methods that aimed to create a *random* scene, we aim to generate a *specific* scene that is conditioned by a given single-view semantic mask. Rather than only concerned about improving the quality of the generated images, our work is more focused on the mapping from the mask to the NeRF-based scene.

Image-to-Image Translation is about converting an image from one source representation, *e.g.*, semantic masks, to another target representation, *e.g.*, photorealistic images. Since its introduction [Isola et al., 2017], progress has been made with regard to higher image resolution [Wang et al., 2018b], multi-modal outputs [Zhu et al., 2017b, Zheng et al., 2019, Choi et al., 2020], unsupervised learning [Zhu et al., 2017a, Lira et al., 2020], *etc*.More recently, Richardson *et al*. [Richardson et al., 2021] tackled this task by learning to map to the style space of a pre-trained StyleGAN, bypassing the limitation caused by pixel-to-pixel correspondence.

In contrast to all mentioned work that aimed to map a semantic mask to an image, ours is focused on mapping to a 3D scene. We also notice that there are some recent approaches targeted at converting semantic masks to 3D scenes. Huang *et al*. [Huang et al., 2020] introduced rendering novel-view photorealistic images from a given semantic mask, by first applying semantic-to-image translation [Park et al., 2019], then converting the single-view image to a 3D scene modelled by multiplane images (MPI) [Zhou et al., 2018]. Hao *et al*. [Hao et al., 2021] proposed to learn a mapping from a semantically-labelled 3D block world to a NeRF-based 3D scene, using a scene-specific setting. Chen *et al*. [Chen et al., 2022] introduced a 3D-aware portrait generator by first mapping the given latent code to a semantic occupancy field (SOF) [Chen and Zhang, 2019] for rendering novel view semantic masks, followed by applying image-to-image translation.

Unlike all mentioned attempts on learning semantic to 3D scene mappings, ours is the first to introduce the single-view semantic to NeRF translation task. Our work differs from theirs in: 1) We do not rely on any separate image-to-image translation stage, resulting in better multi-view consistency; 2) We do not require multi-view semantic masks for both training and testing phases, easing the data collection effort; 3) We pinpoint a solution for creating pseudo labels and demonstrate reasonable results beyond the human face domain.

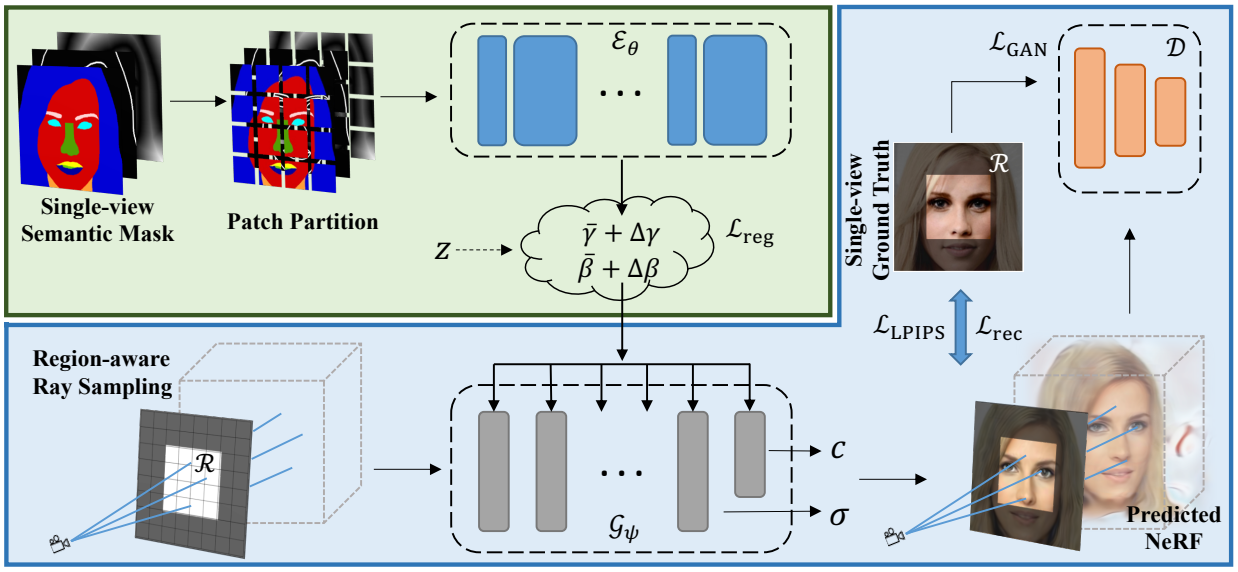

Figure 2: Architecture of the Sem2NeRF framework. It aims to convert a single-view semantic mask to a 3D scene represented by NeRF. Specifically, a given semantic mask will be partitioned into patches, which will be further encoded by a patch-based encoder $\mathcal{E}_\theta$ into a latent style code (γ, β) of a pretrained NeRF-based 3D generator $\mathcal{G}_\psi$. A region $\mathcal{R}$ will be randomly sampled to enforce awareness of differences among regions. And an optional latent vector $\mathbf{z}$ is included to enable multi-modal synthesis

3 Methodology

As shown in Figure 1, our main goal is to train a Semantic-to-NeRF translation network $\Phi_{s\to\mathcal{V}}$, such that when presented with a single-view 2D semantic mask $\mathbf{s}$, it generates the corresponding NeRF representation $\mathcal{V}$, which can then be used to render realistic 3D-consistent images. This task is conceptually similar to the conventional semantic-to-image setting, except that here we opt to go beyond 2D image translation, and deal with the novel *controllable 3D translation.* More importantly, we can freely change the 3D content by *simply modifying the corresponding content in a single-view 2D semantic mask*.

In order to learn such a framework without enough supervision for arbitrary view appearances, we observed that 3D information can be learned from large image collections [Kanazawa et al., 2018, Chan et al., 2021a,b]. Therefore, our *key motivational insight* is this: instead of directly training $\Phi_{s\to\mathcal{V}}$ using *single-view* semantic-image pairs $(\mathbf{s}, \mathbf{I})$ (like current methods for 2D semantic-to-image translation [Wang et al., 2018b, Park et al., 2019]), we will train it as a two-stage pipeline shown in Figure 2. In this pipeline, (A) we utilize a pre-trained 3D generator (lower portion $\mathcal{G}_\psi$) that learns 3D shape and appearance information from a large set of collected images; (B) we pose this challenging task as a *3D inversion* problem, where our main target is to design a front-end encoder (upper portion $\mathcal{E}_\theta$) that maps the semantic mask into the generator's latent space accurately.

Note that the two training stages are executed independently and can be separately implemented with different state-of-the-art frameworks. There are at least two unique benefits of breaking down the entire controllable 3D translation into two-stages: 1) The training does *not* require copious views of semantic-image pairs for each instance, which are difficult to collect, or even impossible in some scenarios; 2) The compartmentalization of the 3D generator and the 2D encoder allows greater agility, where the 3D information can be previously learned on various tasks with a large collection of images and then be freely plugged into the 3D inversion pipeline.

3.1 3D Decoder with Region-Aware Ray Sampling

Preliminaries on Neural Radiance Fields (NeRF). We first provide some preliminaries on NeRF before discussing how we exploit it for Sem2NeRF. NeRF [Mildenhall et al., 2020] is one kind of implicit functions that represents a continuous 3D scene, which has achieved great successes in modeling 3D shape and appearance. A NeRF is a neural network that maps a 3D location $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{S}^2$ to a spatially varying volume density σ and a view-dependent emitted color $\mathbf{c} = (r, g, b)$. NeRFs trained on natural images are able to continuously render realistic images at arbitrary views. In particular, it requires to use the volume rendering [Levoy, 1990], which computes the

following integral to obtain the color of a pixel:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \text{where } T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds), \tag{1}$$

where $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ is the ray casting from the virtual camera located at $\boldsymbol{o}$, bounded by near t_n and far t_f, and $T(t)$ represents the accumulated transmittance of the ray traveling from t_n to t. The integral $C(\mathbf{r})$ is further implemented with a hierarchical volume sampling strategy [Levoy, 1990, Mildenhall et al., 2020], resulting in the optimization of a "coarse" network followed by a "fine" network.

NeRF-based Generator. In this work, our study is mainly based on a representative NeRF-based generator, π-GAN [Chan et al., 2021a], which learns 3D representation using only 2D supervision. Inspired by StyleGAN2 [Karras et al., 2020], the architecture of π-GAN is mainly composed of two parts, a mapping network $\mathcal{F} : \mathcal{Z} \rightarrow \mathcal{W}$ that maps a latent vector z in the input latent space $\mathcal{Z}$ to an intermediate latent vector $w \rightarrow \mathcal{W}$, and a SIREN-based [Sitzmann et al., 2020] synthesis network that maps w to the NeRF representation $\mathcal{V}$ that supports rendering 3D-consistent images from arbitrary camera poses.

Our proposed Sem2NeRF framework can use various NeRF-based generators. Here, we choose π-GAN as the main decoder in our architecture for two main reasons. Firstly, among all existing *published* works related to NeRF-based generators, π-GAN achieves state-of-the-art performance in terms of rendered image quality and their underlying 3D consistency. Secondly and more importantly, similar to StyleGAN, the FiLM [Perez et al., 2018] conditioning used by π-GAN enables layer-wise control over the decoder and the mapping network decouples some high-level attributes, making it easier to perform *3D inversion* on top of NeRF, *i.e.*searching for the desired latent code w that best reconstructs an ideal target. The similar observation has been previously explored in the latest 2D GAN inversion [Collins et al., 2020, Karras et al., 2020, Richardson et al., 2021].

Region-Aware Ray Sampling. While π-GAN already provides high-quality view-consistent rendered images, our main goal is to accurately restore the NeRF from a single-view semantic mask, and even freely edit the 3D content via such a map. To achieve this, *the network should be sensitive to local small modifications.* However, this is not supported in the original π-GAN, which is trained on each entire image with a global perception. It stores scene-specific information in a latent code, which is shared across all points that are bounded by the rendering volume. As a result, a small change in an original latent code will easily cause a global modification in generation. This may not impact pure 3D generation, for which only the quality of global shape and appearance is paramount, but it has a large negative effect on recreating a 3D representation that accurately matches the corresponding semantic mask.

To mitigate this issue, we adopt a region-based ray sampling pattern [Schwarz et al., 2020, Liu et al., 2022] in the π-GAN decoder, that *attempts to encourage latent codes to represent local regions at different scales and locations*. Suppose the rendered image $\mathbf{I}$ with a target size $h \times w$, a local region $\mathcal{R}$ used for training is randomly sampled as

$$\mathcal{R}(\alpha, (\Delta h, \Delta w)) = \{(\alpha h + \Delta h, \alpha w + \Delta w)\}, \tag{2}$$

where $(\alpha h + \Delta h, \alpha w + \Delta w)$ denotes the sampling coordinates of rays, with $\alpha \in (0, 1]$ being the scaling factor and $(\Delta h \in [0, (1-\alpha)h], \Delta w \in [0, (1-\alpha)w])$ being the translation factor. To obtain such training pairs between the NeRF rendered output and the local ground truth, we sample the original whole image using the same region coordinates $\mathcal{R}$ with bilinear interpolation. This region-aware strategy leads to large improvements on conditional generation as we shown in the experiments.

3.2 3D Inversion Using Region-Aware 2D Encoder

3D Inversion. In order to inversely map a semantic mask $\mathbf{s}$ into the $\mathcal{W}$ latent space of the 3D generator $\mathcal{G}_\psi$ by an encoder $\mathcal{E}_\theta$, with respective parameters ψ and θ, we train $\mathcal{E}_\theta$ to minimize the reconstruction error between ground truth image $\mathbf{I}$ and output image $\hat{\mathbf{I}}$. Specifically, the Semantic-to-NeRF translation represents the mapping

$$\Phi_{\mathbf{s}\rightarrow\mathcal{V}}(\mathbf{x}, \mathbf{d}, \mathbf{z}; \mathbf{s}) = \mathcal{G}_\psi(\mathbf{x}, \mathbf{d}, \mathbf{z}; \mathcal{E}_\theta(\mathbf{s})) = \mathcal{V}(\sigma, \mathbf{c}) \tag{3}$$

where $\mathbf{x}, \mathbf{d}$ denotes point position and ray direction, while the derived density σ and color $\boldsymbol{c}$ can be used to calculate the corresponding pixel value via volume rendering as in Eq. (1). For *controllable* 3D generation, $\mathbf{s}$ is the input single-view semantic mask, embedded into $\mathcal{W}$ space to control the generated 3D content, while we also enable multi-modal synthesis by adding another latent vector $\mathbf{z}$ to model the generated appearance. Note that $\mathbf{s}$ only comes in a single view, which is not necessary the same as the output viewing direction. In short, *we use only single-view semantic-image pairs* $(\mathbf{s}, \mathbf{I})$ *for the Sem2NeRF training*, as the 3D view-consistent information has been captured by the *fixed* pre-trained 3D generator $\mathcal{G}_\psi$. Hence, we focus only on training the encoder network $\mathcal{E}_\theta$ to learn the posterior distribution $q(w|\mathbf{s})$ for 3D inversion.

Region-Aware 2D Encoder. A simple way for 3D inversion is to directly apply an existing 2D GAN inversion framework. However, this straightforward idea does *not* work well as we originally discovered when using the state-of-the-art pSp encoder [Richardson et al., 2021] in our setting, especially for small but perceptually important regions, such as eyes. Our conjecture is that the conventional CNN-based architecture integrates the neighboring information via overaggressive filtering, resulting in heavy loss of small details [Zhang, 2019].

To mitigate this issue, we also deploy a region-aware learning strategy in the 2D encoder, which is inspired by the latest patch-based methods [Dosovitskiy et al., 2020, Zheng et al., 2021] that capture information in every patch with equal possibility. In other words, when we directly extract features from local patches, it will be *more sensitive to the semantic variation within each patch*, which can ameliorate the problem of imbalanced semantic distribution within an image. In particular, we adopt the Swin Transformer [Liu et al., 2021] as the encoder architecture. To embed the semantic mask $\mathbf{s}$ into the $\mathcal{W}$ latent space of the pre-trained 3D generator, we replace the final classification output size with the size of the latent vectors $\mathbf{w}$. Besides, to further stabilize the inversion training, we apply the "delta latent space" strategy of pSp [Richardson et al., 2021] into our framework. The learned latent codes for the pre-trained decoder then become

$$\gamma = \overline{\gamma} + \Delta\gamma, \beta = \overline{\beta} + \Delta\beta, \tag{4}$$

where γ and β represent the embedded vectors for the $\mathcal{W}$ latent space, *i.e.*, frequency and phase shift of π-GAN, respectively; $\Delta\gamma$ and $\Delta\beta$ are the outputs of the proposed encoder $\mathcal{E}_\theta$, while $\overline{\gamma}$ and $\overline{\beta}$ are the average latent codes extracted by the pre-trained π-GAN mapping network $\mathcal{F} : \mathcal{Z} \rightarrow \mathcal{W}$.

Additional Inputs for the 2D Encoder. As mentioned, a semantic mask contains sparse information, where the changing of small regions may be imperceptible to the network, making the semantic-based controllable 3D editing very challenging. Considering that editing a semantic mask only effectively alters the boundaries between different semantic labels, we conjecture that explicitly augmenting the semantic input with boundary information will be useful for semantic editing. Therefore, we concatenate the semantic mask input with contours and distance field representations [Chen and Hays, 2018] for the region-aware encoder. These additional inputs further improve the semantic editing performance considerably as shown in the experiments. Note that contours and distance field representations are both directly calculated from the semantic masks (refer to Section A.1 for more technical details), which do *not involve any extra labels*.

3.3 Training Loss Functions

During the training phase, we use the single-view semantic mask $\mathbf{s}$, the corresponding viewing direction d_s, and the paired ground truth RGB image $\mathbf{I}$. Similar to Semantic-to-Image translation, we start by applying a pixel-level reconstruction loss,

$$\mathcal{L}_{\text{rec}}(\mathbf{I}, \mathbf{s}, d_s) = \left\| \mathbf{I} - \mathcal{G}_\psi(\mathcal{E}_\theta(\mathbf{s}), d_s) \right\|_2, \tag{5}$$

where $\mathcal{E}_\theta(\mathbf{s})$ denotes the latent codes mapped from $\mathbf{s}$ via the region-aware encoder $\mathcal{E}_\theta(\cdot)$, while $\mathcal{G}_\psi(\mathcal{E}_\theta(\mathbf{s}), d_s)$ represents the generated image rendered from direction d_s via the decoder $\mathcal{G}_\psi(\cdot)$. Unless otherwise specified, the aforementioned region-aware sampling strategy is applied to $\mathcal{G}_\psi$ and $\mathbf{I}$ before calculating any losses.

To further enforce the feature-level similarity between the generated image and the ground truth, the LPIPS loss [Zhang et al., 2018] is leveraged,

$$\mathcal{L}_{\text{LPIPS}}(\mathbf{I}, \mathbf{s}, d_s) = \left\| \mathcal{F}(\mathbf{I}) - \mathcal{F}(\mathcal{G}_\psi(\mathcal{E}_\theta(\mathbf{s}), d_s)) \right\|_2, \tag{6}$$

where $\mathcal{F}(\cdot)$ refers to the pre-trained feature extraction network.

Inspired by the truncation trick proposed in StyleGAN, we further encourage the decoder latent codes γ, β to be close to the average codes $\overline{\gamma}$, $\overline{\beta}$, which is achieved by regularizing the encoder with

$$\mathcal{L}_{\text{reg}}(\mathbf{s}) = \left\| \mathcal{E}_\theta(\mathbf{s}) \right\|_2. \tag{7}$$

To improve image quality, especially for novel views, we further apply a non-saturating GAN loss with R1 regularization [Mescheder et al., 2018],

$$\begin{aligned} \mathcal{L}_{\text{GAN}}(\mathbf{I}, \mathbf{s}, d) = f(\mathcal{D}(\mathcal{G}_\psi(\mathcal{E}_\theta(\mathbf{I}), d))) + f(-\mathcal{D}(\mathbf{I})) + \lambda_{\text{R1}} \left| \nabla \mathcal{D}(\mathbf{I}) \right|^2, \\ \text{where } f(u) = -\log(1 + \exp(-u)). \end{aligned} \tag{8}$$

Here $\mathcal{D}(\cdot)$ is a patch discriminator [Isola et al., 2017], aligned with our region-aware learning strategy for the decoder, and λ_{R1} is a hyperparameter that is set to 10. Note that here the viewing direction d is not required to be the same as the input semantic viewing direction d_s, and we randomly sample this viewing direction from a known distribution, *i.e.* Gaussian, following the settings of π-GAN [Chan et al., 2021a].

Finally, the overall training objective for our framework is a weighted combination of the above loss functions as

$$\mathcal{L}_{\text{Sem2NeRF}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}. \tag{9}$$

3.4 Model Inference

For inference, our model takes as input a 2D single-view semantic mask, while d_s is optional, required only when rendering an image with the same viewing direction as the semantic mask. Different from the training phase, during inference the rays are cast to cover the whole image plane, rather than a local region.

Multi-View Generation. Since the employed decoder is a NeRF-based generator, it inherently supports novel view generation. Specifically, given a semantic mask $\mathbf{s}$, it will first be mapped as an embedded vector in the $\mathcal{W}$ latent space that controls the "content" of the NeRF-based generator, whereupon a novel view image can then be generated by volume rendering the NeRF from an arbitrary viewing direction.

Multi-Modal Synthesis. Similar to the diversified mapping in semantic-to-image [Park et al., 2019], ideally a single semantic mask should be translated into multiple NeRFs consistent to it. As π-GAN uses FiLM conditioning to control the "style" as in StyleGAN, it can achieve multi-modal synthesis by simply manipulating specific layers of the style codes, due to the disentanglement of its latent space. Therefore, our Sem2NeRF framework inherently supports multi-modal synthesis in inference, without requiring any special customization in training. In practice, we additionally pass a random-sampled vector to the pre-trained π-GAN noise mapping module to obtain corresponding latent style codes $\boldsymbol{z}$. Style mixing [Richardson et al., 2021, Karras et al., 2019] is then performed between $\boldsymbol{z}$ and $\mathcal{E}_\theta(\mathbf{s})$ to yield multi-modal outcomes.

4 Experiments

4.1 Settings

Datasets. To achieve Semantic-to-NeRF translation, we assume the training data to have single-view registered semantic masks and images, with the corresponding viewing directions. Two datasets were used for evaluation in our experiments. **CelebAMask-HQ** [Lee et al., 2020] contains images from CelebA-HQ [Liu et al., 2015, Karras et al., 2017], manually-labelled 19-class semantic masks, and head poses. We merged the left-right labels of symmetric parts, *i.e.*, eyes, eyebrows and ears, into one label per part. The dataset was randomly partitioned into training set with 28,000 samples and test set with 2,000 samples. **CatMask** is built using π-GAN and DatasetGAN [Zhang et al., 2021] to further demonstrate the potential of the Semantic-to-NeRF task and our Sem2NeRF (see Section A.2 for more technical details). It comes with 30,000 cat faces, 6-class semantic masks, and pseudo-ground-truth viewing directions. It uses the same training partition as CelebAMask-HQ.

Baselines. We identified the following three methods as baselines for comparison in our introduced Semantic-to-NeRF task. **SofGAN** [Chen et al., 2022] is an image translation approach. For a given single-view mask, we first apply inversion via iterative optimizations to find the corresponding latent vector for the preceding SOF [Chen and Zhang, 2019] network, which can generate novel view semantic masks for further image-to-image mapping. Note that SofGAN requires training data to have high-quality multi-view semantic masks, which is not available nor needed in our task. We compare with SofGAN using the released codes and models. **pix2pixHD** [Wang et al., 2018b] is an image translation approach. We adopt it with general GAN-inversion techniques [Chan et al., 2021a, Karras et al., 2020]. For a given mask, it is first mapped to a photorealistic image via pix2pixHD, which will then be mapped to the corresponding latent code in π-GAN via GAN-inversion. With the recovered codes, multi-view images can be directly obtained using π-GAN. **pSp** [Richardson et al., 2021] is an image translation approach that is designed for encoding into StyleGAN2 [Karras et al., 2020]. We adapted it by using its ResNet [He et al., 2016]-based pSp encoder to replace the π-GAN mapping network, and we further trained the network with objective functions used by pSp.

Evaluation Metrics. We show qualitative results by rendering images with different viewing directions and FOV (Field of View). We also report Frechet Inception Distance (FID) [Heusel et al., 2017] and Inception Score (IS) [Salimans et al., 2016] using Inception-v3 [Szegedy et al., 2016] over the test sets. Average running time and model sizes are also compared.

Implementation Details. Swin-T is used in all experiments with input resolution 224×224. For the decoder, the size of local region $\mathcal{R}$ is set to 128×128. The step size of each ray is set to 28. Other miscellaneous settings of the pre-trained decoder, *e.g.*, ray depth ranges, are kept unchanged. Hyper-parameters in Eq. (9) are set as λ_{rec}=1, λ_{LPIPS}=0.8, λ_{reg}=0.005, λ_{GAN}=0.08. The implementation is done in PyTorch [Paszke et al., 2019]. All baselines and ablation models used similar experiment settings. More details are provided in Section A.3.

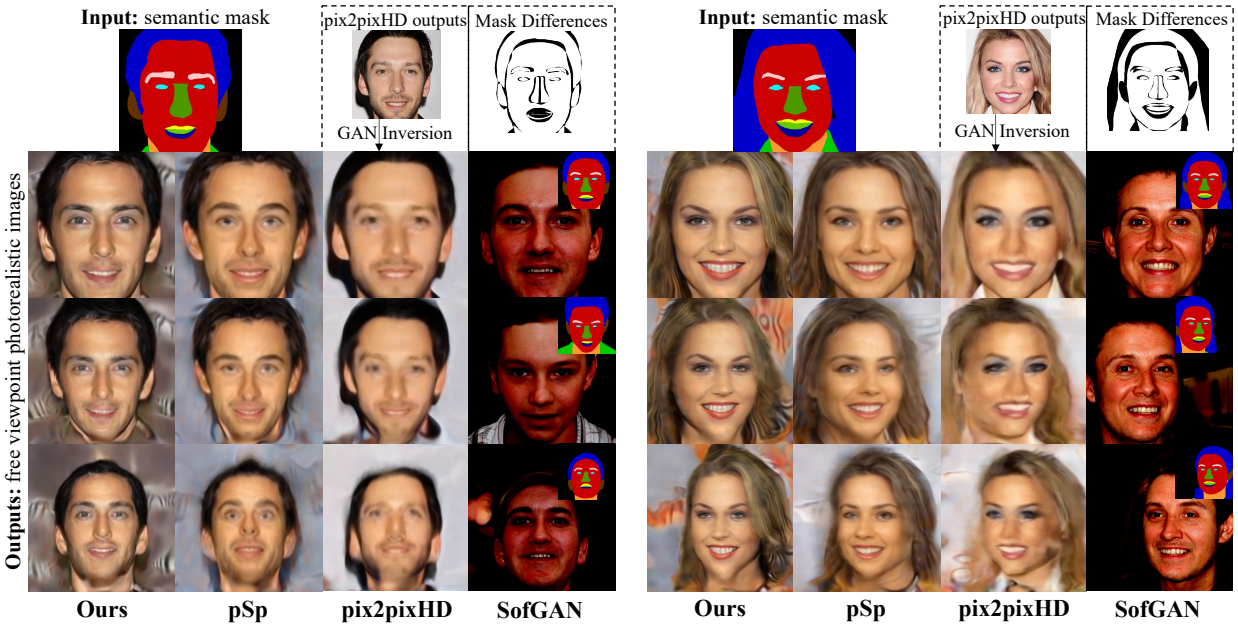

Figure 3: Comparisons on CelebAMask. Images at each column are generated by the corresponding models mentioned at the bottom. Only SofGAN requires generation of multi-view semantic masks, shown at the top right corners of related images

4.2 Results

Comparisons on CelebAMask-HQ. As shown in Figure 3, compared to all other baseline models, **Sem2NeRF (1st&5th columns)** achieved the best performance on both mapping accuracy and multi-view consistency. **pSp (2nd&6th columns)** generated images with lower quality compared to ours, especially for novel views, mainly because our model is designed with a region-aware learning strategy and a GAN loss for random-posed images during training. The CNN-based encoder also failed to capture fine-grained details, *e.g.*, eyebrow shapes for the left face. Our method and the inversion-based pix2pixHD were better in matching semantics compared to pSp. **pix2pixHD (3rd&7th columns)** can map semantic masks to high quality images in the same viewpoint (top row), but does not generate novel views well. Basic GAN-inversion is not an efficient or easy way to find the desired latent codes, since the current 3D generative models are still quite immature. Even though images with the same viewing direction as the masks are reasonable, those novel view outputs contain artifacts. **SofGAN (4th&8th columns)** generated each single-view image with good quality; however, its results do not match with the given mask and lacked 3D consistency. The reason is that it is hard to map the given semantic mask to the desired latent codes of its semantic generator (SOF Net), whose sampling space is relatively small due to the lack of training data (only 122 subjects). The recovered mask did not match well with the given mask (top row). Besides, although the semantic masks show good multi-view consistency (top right corner of each image), conducting semantic-to-image mapping separately for each viewpoint does not guarantee that the consistency will be retained, since a semantic mask hardly contains any texture information and is geometrically ambiguous.

Quantitative results are give in Table 1. It can be seen that our Sem2NeRF method achieves the best performance, significantly outperforming the two baselines in both FID and IS scores. Note that we did not quantitatively compare single view image quality with SofGAN, considering that SofGAN for Semantic-to-NeRF is limited by its mask inversion quality and multi-view consistency, both of which cannot be measured by FID or IS scores. We also notice that scores of all models are lower than expected. The main reason is that π-GAN is initially trained on CelebA, but due to the requirement of semantic masks, our task conducted experiments using CelebA-HQ. The domain gap between CelebA and CelebA-HQ reduced the FID scores dramatically. Besides, our model also sees advantages in terms of running time and model size.

Mask Editing. As depicted in Figure 4, our framework supports editing of 3D scenes by simply changing the given semantic mask, and is applicable to both labels associated with large regions, *e.g.*, hair, as well as small regions, *e.g.*, eyes, nose, mouth. This is not trivial since the semantic mask is not directly leveraged to control the 3D scene at the

Table 1: Quantitative comparisons on CelebAMask @ 128×128

	FID ↓	IS ↑	Runtime(s) ↓	# Params(M) ↓
pix2pixHD [Wang et al., 2018b] (with inversion)	67.32	1.72	161.59±0.859	∼184.24
pSp [Richardson et al., 2021]	55.56	1.74	0.25±0.004	∼138.27
Sem2NeRF(Ours)	**41.52**	**2.03**	**0.18±0.003**	**∼32.01**

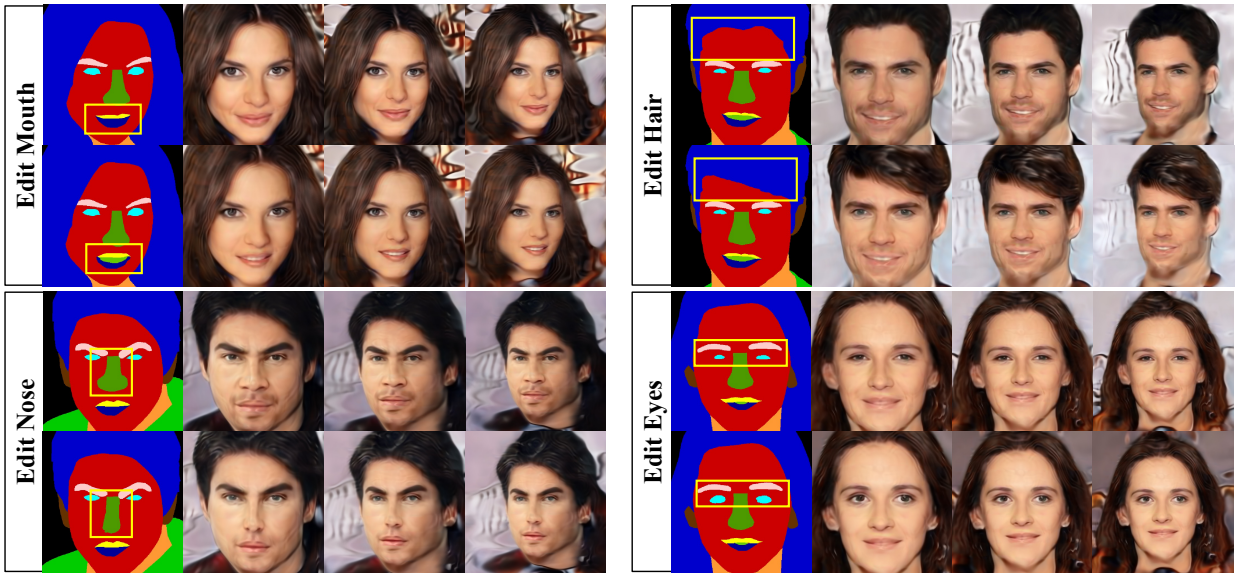

Figure 4: Editing 3D scenes by changing single-view semantic masks. Three viewpoints are shown for better comparison in each group

Figure 5: Multi-modal synthesis. Styles are linearly blended from left to right. Three viewpoints are provided from top to bottom

pixel level (if even possible), but is instead encoded into a sparse latent code, which may fail to preserve fine-grain editing. We address this challenge via the region-aware learning strategy.

Multi-Modal Synthesis. Inherent in the π-GAN decoder, Sem2NeRF also supports multi-modal synthesis by simply changing the last few layers (normally last 1 or 2) of the style codes. As shown in Figure 5, we randomly sampled two style codes, and applied linear blending to continuously change the general styles of the 3D scenes generated by the given masks.

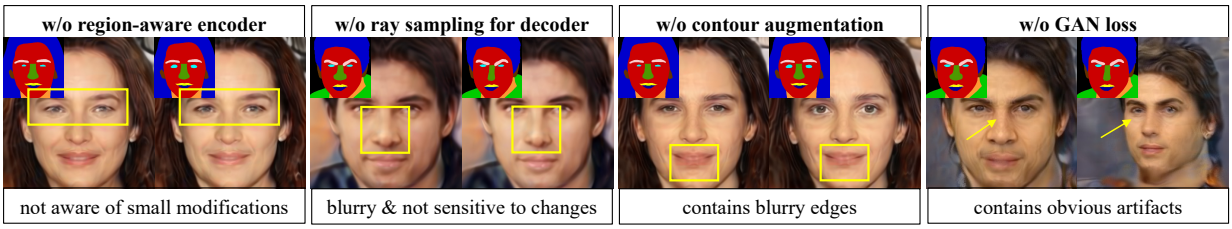

Figure 6: Results of ablation studies. Each group (two views) is generated by a model without the component mentioned at the top. Main issues are described at the bottom

Figure 7: Results on CatMask. Left part compares results of changing eyes shape. Right part showcases results of style linear blending (in zigzag order)

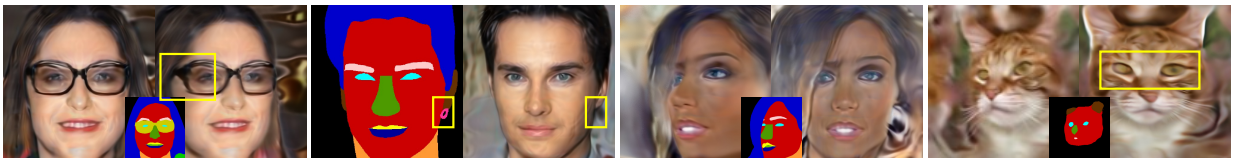

Figure 8: Challenging cases of Sem2NeRF on Semantic-to-NeRF translation

Ablation Studies. To further evaluate the efficacy of Sem2NeRF, we designed four ablation models, including 1) without region-aware encoder $\Phi_{\text{no\_RE}}$, where the Swin-T encoder is replaced by the pSp encoder; 2) without region-aware decoder $\Phi_{\text{no\_RD}}$, where the region-aware ray sampling strategy is discarded; 3) without input augmentation $\Phi_{\text{no\_IA}}$, where contours and distance field representations are removed from the input; and 4) without random-pose GAN loss $\Phi_{\text{no\_GAN}}$, where both Eq. (8) and the discriminator are removed.

As shown in Figure 6, compared to the full model (Figure 4), $\Phi_{\text{no\_RE}}$ (1st group) is not sensitive to changes in small regions, *i.e.*, eyes, mainly because the CNN-based encoder tends to ignore small changes. $\Phi_{\text{no\_RD}}$ (2nd group) shows similar pattern (nose region) as the latent codes are not trained to be region-aware. It also has lower image quality, because the region-aware strategy enables denser sampling. $\Phi_{\text{no\_IA}}$ (3rd group) achieves comparable performance but with blurry edges for some regions, *e.g.*, mouth. This is because both contour and distance field representation help highlight the border information. Finally, images obtained by $\Phi_{\text{no\_GAN}}$ (4th group) have more artifacts in both views, demonstrating that GAN loss is important for improving the image quality of different poses.

Experiments beyond Human Faces. The introduced task can easily go beyond the human face domain by leveraging state-of-the-art weakly supervised semantic segmentation model to create pseudo labels. In this work, we present a Cat face example. Experimental results are shown in Figure 7. Even when training with noisy pseudo labels, Sem2NeRF is robust enough to generate plausible results. For a given cat semantic mask, our model can map it to a 3D scene and render cat faces from arbitrary viewpoints, including different viewing directions (left part), and different FOV (right part). It also allows changing the 3D scenes by editing the single-view semantic masks, *e.g.*, changing the eye shape (left two rows). Multi-modal synthesis is also supported (right part in zigzag order).

Challenging Cases. Although Sem2NeRF addresses the Semantic-to-NeRF task in most cases, its advantages rely on an assumption, namely the generative capability of the pretrained decoder. We show some challenging cases in Figure 8. Accessories may have the wrong geometric shape (glasses in 1st case), or fail to render (earring in 2nd case), while masks with extreme poses might be converted to 3D scenes with abnormal texture or distorted contents (last two cases).

5 Conclusions

We have presented an initial step of extending the 2D image-to-image task to the 3D space, and introduced a new task called Semantic-to-NeRF translation. It aims to reconstruct a 3D scene represented by a NeRF, by taking as input only one single-view semantic mask. We further proposed Sem2NeRF model, which addresses the task via encoding the semantic mask into the latent space of a pretrained 3D generative model. More importantly, we intriguingly found the importance of regional awareness for this new task, and tamed Sem2NeRF with a region-aware learning strategy. We demonstrated the capability of Sem2NeRF regarding multi-view consistency, mask editing and multi-modal synthesis on two benchmark datasets, and showcased the superiority of our framework compared to three strong baselines. Future work will include adding more scenarios to the new task, and supporting changing styles for specific regions.

References

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

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

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

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018a.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021a.

Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. *arXiv preprint arXiv:2110.11323*, 2021b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2014.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021a.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. *arXiv preprint arXiv:2112.07945*, 2021b.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021.

Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. Agilegan: stylizing portraits by inversion-consistent transfer learning. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.

Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG)*, 41(1):1–26, 2022.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018b.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1323–1330. IEEE, 2021.

Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. Semantic-aware implicit neural audio-driven video portrait generation. *arXiv preprint arXiv:2201.07786*, 2022.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

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.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017b.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

Wallace Lira, Johannes Merz, Daniel Ritchie, Daniel Cohen-Or, and Hao Zhang. Ganhopper: Multi-hop gan for unsupervised image-to-image translation. In *European conference on computer vision*, pages 363–379. Springer, 2020.

Hsin-Ping Huang, Hung-Yu Tseng, Hsin-Ying Lee, and Jia-Bin Huang. Semantic view synthesis. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14072–14082, 2021.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

Marc Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics (TOG)*, 9(3):245–261, 1990.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020.

Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Tfill: Image completion via a transformer-based architecture. *arXiv preprint arXiv:2104.00845*, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

A Additional Technical Details

A.1 Builders for Additional Inputs

As mentioned in Section 3.2, additional inputs, *i.e.*, contour and distance field representation, are provided for the encoder of Sem2NeRF to further highlight the boundary information. In this subsection, we will detail how to build these data using the python package "cv2"[1]. Note that both of them are directly calculated from the semantic mask, without involving any extra labels. And each builder function can be done in $1 \sim 2$ milliseconds.

Contour is generally represented as a curve, joining all the continuous points that share the same intensity. It is widely used as a tool to help shape analysis, object detection, *etc*.. In our implementation, we build the contour from the given one-hot encoded semantic mask. Main python codes are given as below. An output example is shown in Figure 9 (middle).

```
# ------------------------------ CONTOUR BUILDER --------------------------------
def binary_masks_to_contour(binary_masks):
    ''' INPUT: binary_masks, semantic mask in one-hot encoding form
        OUTPUT: contour, a contour map of the given semantic mask '''
    # initialize a black canvas
    mask = numpy.zeros((512, 512, 3), dtype=numpy.uint8)
    # find contours for each label
    for binary_mask in binary_masks:
        cnts = cv2.findContours(binary_mask, cv2.RETR_EXTERNAL,
                                cv2.CHAIN_APPROX_SIMPLE)
        cnts = cnts[0] if len(cnts) == 2 else cnts[1]
        for c in cnts:
            # draw contour with white color on the canvas
            cv2.drawContours(mask, [c], -1, (255, 255, 255), thickness=3)
    contour = cv2.cvtColor(mask, cv2.COLOR_BGR2GRAY)
    return contour
# ------------------------------------ END ----------------------------------
```

Distance field representation is a dense representation extracted from binary image via distance transformation. In the distance field, the grey intensity of each pixel indicates its distance to the nearest boundary. An unsigned Euclidean distance field representation is adopted in our experiments. Main python codes are given as below. An output example is shown in Figure 9 (right).

```
# -------------------- DISTANCE FIELD REPRESENTATION BUILDER --------------------
def contour_to_dist_filed(contour):
    ''' INPUT: contour, contour of the semantic mask
        OUTPUT: dist_field, distance filed representation '''
    # invert background and foreground of contour to match the setting
    invert_contour = cv2.bitwise_not(contour)
    dist_field = cv2.distanceTransform(invert_contour, cv2.DIST_L2, 3)
    # normalize the distance filed to [0, 1]
    cv2.normalize(dist_field, dist_field, 0, 1.0, cv2.NORM_MINMAX)
    dist_field = dist_field * 255.
    return dist_field
# ------------------------------------ END ----------------------------------
```

A.2 CatMask Dataset Rendering

To better demonstrate the introduced Semantic-to-NeRF translation task and evaluate the proposed Sem2NeRF framework, we develop a general solution to create datasets with pseudo labels using minimal human effort. In this work, we present an example by rendering a cat faces dataset, termed CatMask, which contains single-view cat faces generated by the pre-trained π-GAN model, pseudo ground-truth viewing directions, and 6-classes semantic masks labelled by DatasetGAN [Zhang et al., 2021].

Cat faces from π-GAN. As mentioned in Section 4.1, we assume the training data to have the viewing directions / poses of the single-view semantic-image pairs. However, different from human face, it is difficult to get the poses for cat faces. Considering that a pretrained π-GAN can generate photorealistic cat faces with given random poses, we choose

[1] https://pypi.org/project/opencv-python/

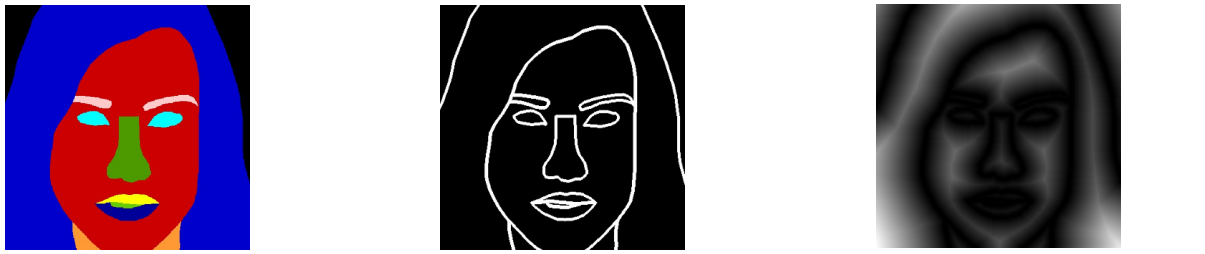

Figure 9: An example of the encoder input. Semantic mask is shown in color for better visualization, while the network takes one-hot encoded mask as input

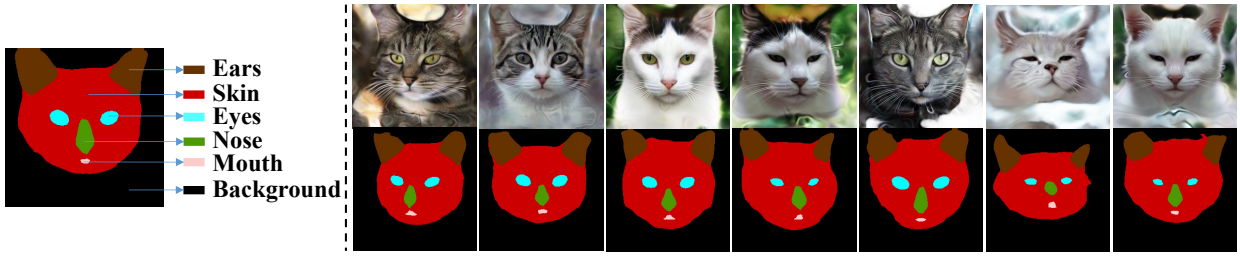

Figure 10: CatMask dataset. Left: label legends of 6 semantic classes. Right: single-view cat faces rendered by π-GAN (top row) and the corresponding semantic masks labelled by DatasetGAN (bottom row). Best view in high quality color image

to generate pseudo data for training our Sem2NeRF. Specifically, we randomly sample 30,000 vectors $\mathbf{z} \in \mathbb{R}^{256}$ from the input distribution of π-GAN to generate 30,000 corresponding cat faces by using the released pretrained model[2]. Each image comes with one viewing direction, randomly sampled from normal distributions, with $X \sim \mathcal{N}(\pi/2,\ 0.3^2)$ for the yaw axis, $X \sim \mathcal{N}(\pi/2,\ 0.1^2)$ for the pitch axis, and 0 for the roll axis. Camera FOV is set to 18 to ensure the generated image covering the full cat face. Ray sampling resolution is set to 512×512, with ray depth range [0.8, 1.2] and ray step size 72. Hierarchical sampling is enabled to improve image quality. We save the rendering viewing direction and the generated images for the CatMask dataset.

Cat semantic by DatasetGAN. A suitable training dataset for Sem2NeRF should be able to be modelled by existing NeRF-based generator, while also comes with semantic labels. However, most datasets used by existing 3D-aware generative models, *e.g.*, cat and car, do not contain component-level semantic masks. We further find out that DatasetGAN can create reasonable semantic masks labels for our task.

DatasetGAN is introduced to automatically build datasets of high-quality semantically segmented images. Specifically, it proposes a MLP-based "Style Interpreter" that can be trained to decode the feature maps of a pretrained StyleGAN model to semantic labels, requiring only very few manually-labeled training samples, *e.g.*, 30 labelled images for cat dataset. In this case, dataset can be automatically built by first randomly sampling images from StyleGAN, following by parsing with the trained style interpreter to obtain corresponding semantic labels.

We use the released[3] pretrained style interpreter for cat to generate a dataset of 10,000 images-semantic pairs. Such a dataset is further leveraged to train a Deeplab-V3 [Chen et al., 2017] model, which takes a cat image as input and outputs the corresponding cat semantic mask. We then use the trained Deeplab-V3 to label our generated CatMask dataset. 6 classes are selected based on the label quality. Label legends and examples of the CatMask dataset are given in Figure 10.

A.3 Additional Implementation Details

Style codes averages ($\overline{\gamma}, \overline{\beta}$). We randomly sample 10,000 vectors $\mathbf{z} \in \mathbb{R}^{256}$ from a standard normal distribution, then feed them through the pretrained mapping network of the origianl π-GAN model, finally average the outputs over the batch dimension to obtain $\overline{\gamma}, \overline{\beta}$.

[2] `https://github.com/marcoamonteiro/pi-GAN`
[3] `https://github.com/nv-tlabs/datasetGAN_release`

Datasets. For CelebAMask-HQ [Lee et al., 2020], both images and semantic masks are loaded as resolution 640×640, then center cropped to 512×512. For CatMask, images and semantic masks are directly loaded as 512×512. Semantic masks are transformed using one-hot encoding, augmented with the aforementioned contours and distance field representations. We do not apply any other data augmentation, *e.g.*, random flip, to avoid harming the pose information.

Training. Patch discriminator with input size 128×128 is adopted in our experiments, using the implementation provided by the GRAF [Schwarz et al., 2020] project[4]. Images are rendered via only the "coarse" network of the decoder, *i.e.*, removing the hierarchical sampling. The sampling range of the scaling factor α of Eq. (2) is initialized as [0.9, 1.0], where the lower bound is exponentially annealed to 0.06 during training. Encoder is initialized with the ImageNet-1K [Deng et al., 2009] pretrained weights, decoder is initialized with π-GAN pretrained weights, while the discriminator is randomly initialized. We freeze the parameters of the decoder, and set the learning rate of the encoder and discriminator to 1×10^{-4} and 2×10^{-5}, respectively. Ranger optimizer[5] is used for both encoder and discriminator. We set the training batch size to 8, and use V100 GPUs to train all related models for 200,000 iterations.

Inference. To render qualitative results, rays are cast with size 512×512 and depth step 72 in the inference. Besides, "fine" network is activated to enable hierarchical sampling. For GAN-inversion used by pix2pixHD as mentioned in Section 4.1, we adopt the implementation from the π-GAN project[6], and set the iteration number to 700 as suggested.

B Additional Visual Results

In the following three pages, we will present additional visual results for the proposed Sem2NeRF regarding free-viewpoint image generation (see Section B.1), semantic mask editing (see Section B.2) and multi-modal synthesis (see Section B.3). Results are demonstrated on both CelebAMask-HQ and CatMask, and they are all best viewed in high quality color image.

[4] `https://github.com/autonomousvision/graf`

[5] `https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer`

[6] `https://github.com/marcoamonteiro/pi-GAN`

B.1 Free-viewpoint Image Generation

Additional visual results of free-viewpoint image generation on CelebAMask-HQ and CatMask datasets are given in Figure 11 and Figure 12, respectively.

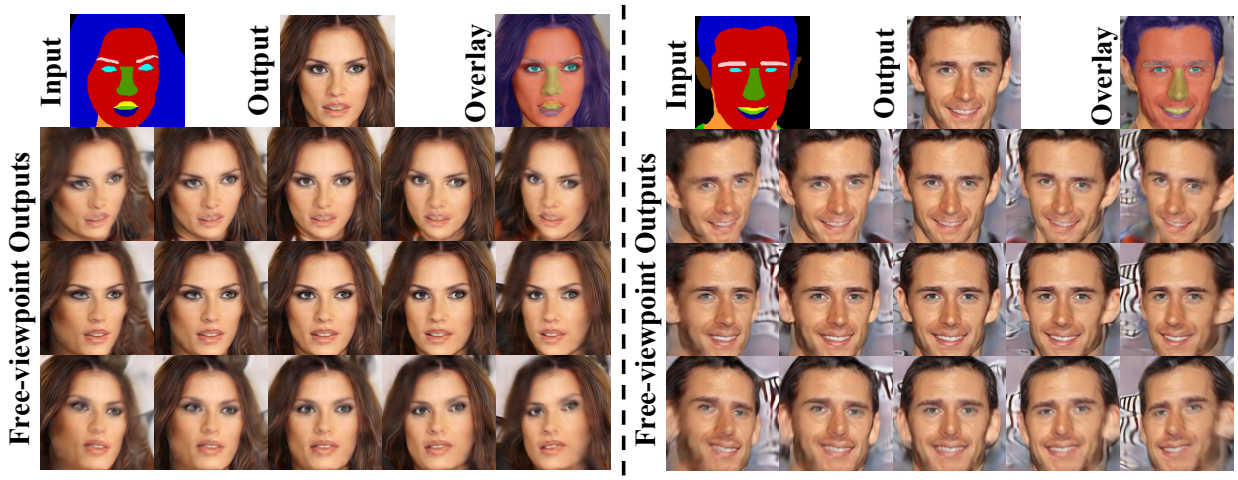

Figure 11: Free-viewpoint image generation on CelebAMask-HQ. "Output" refers to the generated image that has the same viewing direction as the "Input", and "Overlay" shows the results of overlaying "Output" with "Input", so as to better demonstrate the mapping accuracy

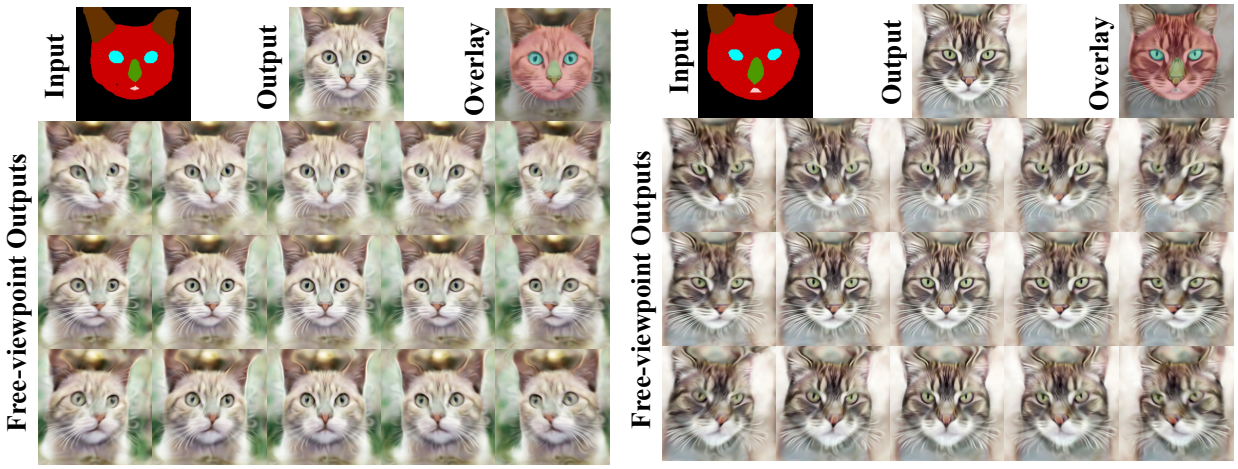

Figure 12: Free-viewpoint image generation on CatMask. "Output" refers to the generated image that has the same viewing direction as the "Input", and "Overlay" shows the results of overlaying "Output" with "Input", so as to better demonstrate the mapping accuracy

B.2 Semantic Mask Editing

Additional visual results of semantic mask editing on CelebAMask-HQ and CatMask datasets are given in Figure 13 and Figure 14, respectively.

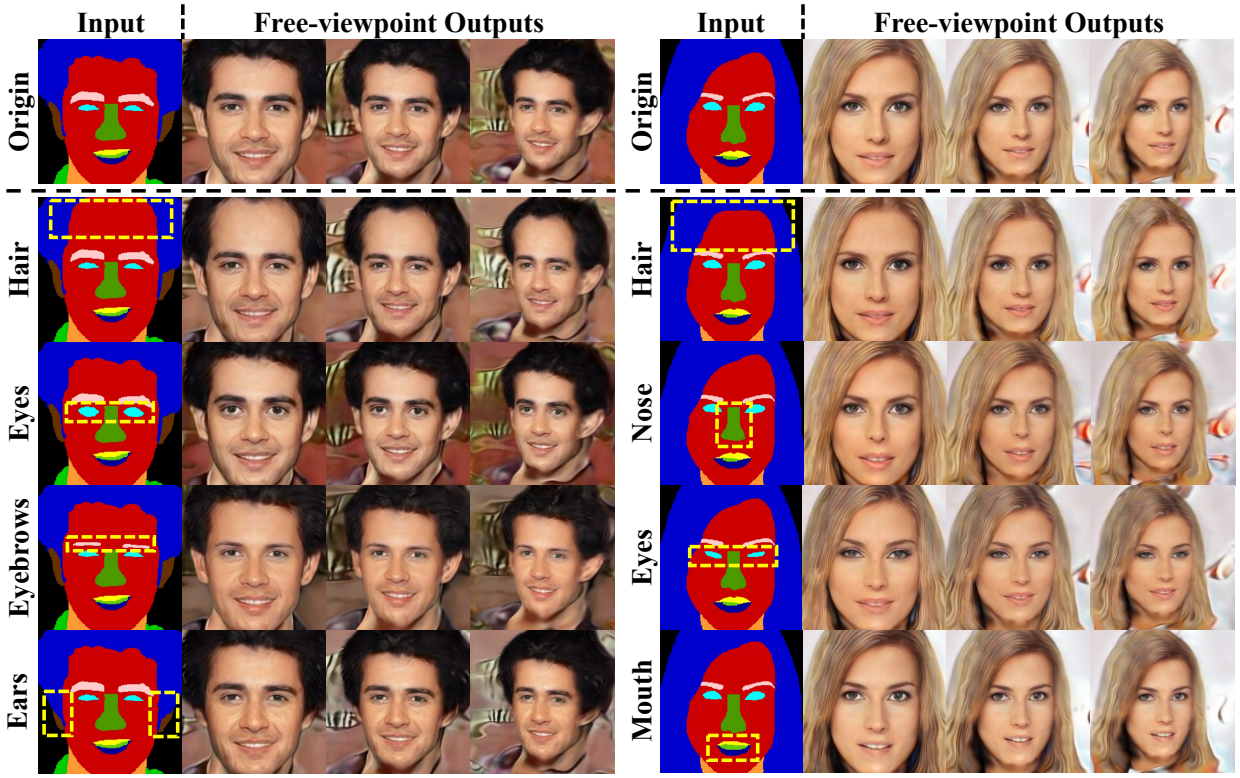

Figure 13: Semantic mask editing on CelebAMask-HQ. The first row shows the results of the original semantic masks, while the following rows give the results of editing the mentioned area, highlighted with yellow-dash box. Three viewpoints are given for each group, with the first one having the same viewing direction as the input

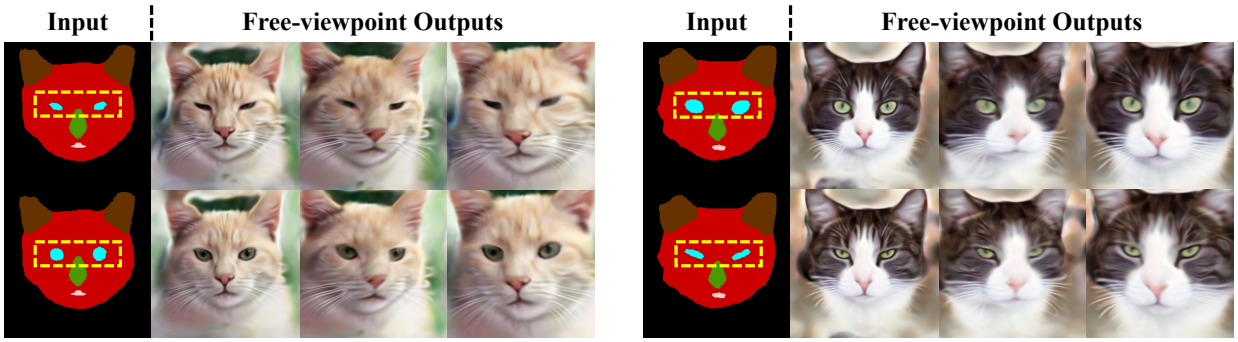

Figure 14: Semantic mask editing on CatMask. Edited regions are highlighted with yellow-dash box. The first viewpoint has the same pose as the input

B.3 Multi-modal Synthesis

Additional visual results regarding multi-modal synthesis on CelebAMask-HQ and CatMask datasets are given in Figure 15 and Figure 16, respectively.

Figure 15: Multi-modal synthesis on CelebAMask-HQ. Styles are linearly blended from left to right. The last viewpoint in each group has the same pose as the input

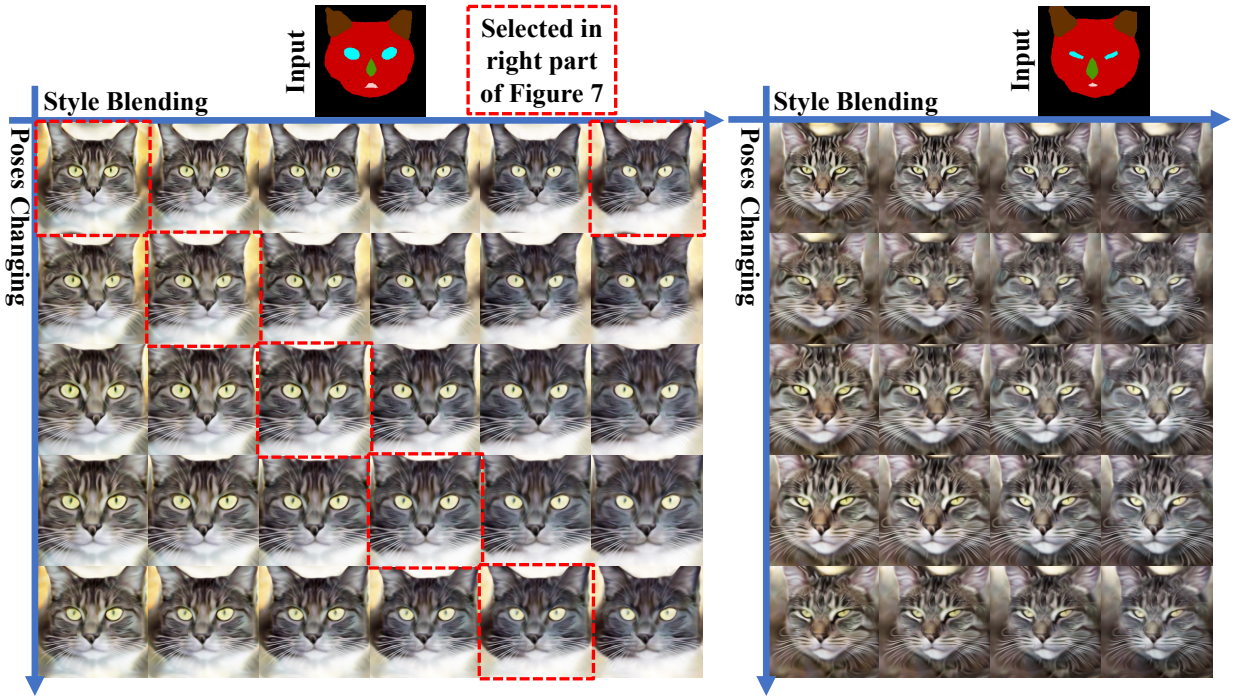

Figure 16: Multi-modal synthesis on CatMask. Left case shows the full version of Figure 7 (right part), where the selected images are highlighted in red-dash border. Images in the first row have the same viewing direction as the input