# ENJOY YOUR EDITING: CONTROLLABLE GANS FOR IMAGE EDITING VIA LATENT SPACE NAVIGATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Controllable semantic image editing enables a user to change entire image attributes with few clicks, e.g., gradually making a summer scene look like it was taken in winter. Classic approaches for this task use a Generative Adversarial Net (GAN) to learn a latent space and suitable latent-space transformations. However, current approaches often suffer from attribute edits which are entangled, global image identity changes, and diminished photo-realism. To address these concerns, we learn multiple attribute transformations simultaneously, we integrate attribute regression into the training of transformation functions, apply a content loss and an adversarial loss that encourage the maintenance of image identity and photo-realism. We propose quantitative evaluation strategies for measuring controllable editing performance, unlike prior work which primarily focuses on qualitative evaluation. Our model permits better control for both single- and multiple-attribute editing, while also preserving image identity and realism during transformation. We provide empirical results for both real and synthetic images, highlighting that our model achieves state-of-the-art performance for targeted image manipulation.

## 1 INTRODUCTION

Semantic image editing is the task of transforming a source image to a target image, while modifying desired semantic attributes, e.g., to make an image taken during summer look like it was captured in winter. The ability to semantically edit images is useful for various real-world tasks, including artistic visualization, design, photo enhancement, and targeted data augmentation. To this end, semantic image editing has two primary goals: (i) providing continuous manipulation of multiple attributes simultaneously and (ii) maintaining the original image's identity as much as possible while ensuring photo-realism.

Existing GAN-based approaches can be categorized roughly into two groups: 1) *image-space editing*, i.e., methods that directly transform one image to another across domains (Choi et al., 2018; 2020; Isola et al., 2017; Lee et al., 2020; Wu et al., 2019; Zhu et al., 2017a;b), usually using variants of generative adversarial nets (GANs) (Goodfellow et al., 2014). These approaches often have high computational cost, and they primarily focus on binary attribute (on/off) changes, rather than providing continuous attribute editing abilities. 2) *latent-space editing*, i.e., methods that focus on discovering latent variable manipulations that permit continuous semantic image edits in a lower-dimensional space, most often using a GAN latent space. Both unsupervised and (self-)supervised latent space editing methods have been proposed. Unsupervised latent-space editing methods (Härkönen et al., 2020; Voynov & Babenko, 2020) are often less effective at providing semantic meaningful directions and all too often change image identity during an edit. Current (self-)supervised methods (Jahanian et al., 2019; Plumerault et al., 2020) are limited to geometric edits such as rotation and scale. To our knowledge, only one supervised approach has been proposed (Shen et al., 2019) – developed to discover semantic latent-space directions for binary attributes. As we show, this method suffers from entangled attributes and often does not preserve image identity during manipulation.

**Contributions.** We propose a latent-space editing framework for semantic image manipulation that fulfills the aforementioned primary goals. For this, we use a GAN and employ a joint sampling strategy, which is in turn trained to edit multiple attributes simultaneously. To disentangle attribute transformations in the latent space of GANs, we integrate a regressor that predicts the attributes that an image exhibits. The regressor also permits precise control of the degree of manipulation, and

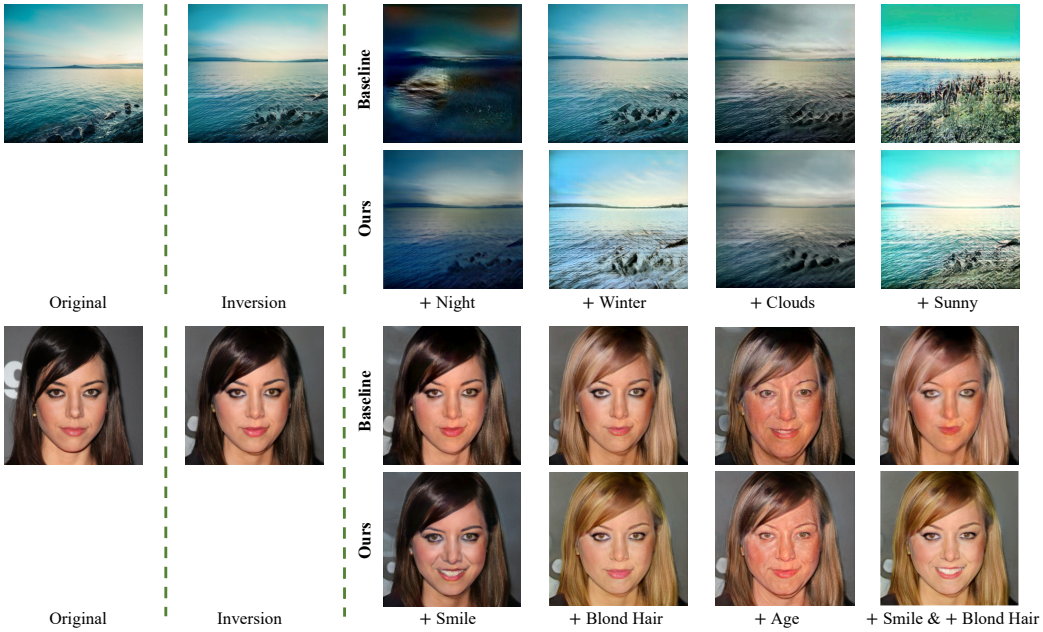


Figure 1: Real image manipulation on scene (top two rows, photo from Flickr) and face (bottom two rows, unseen image from CelebA-HQ) using pretrained StyleGAN2 (Karras et al., 2019b): We reconstruct the real images (col.1) using Abdal et al. (2019) by finding a latent vector with the best inversion result (col.2) on StyleGAN2. After that, we transform the latent vectors for single- and multiple-attribute manipulations (col.3-6). Note that unlike ours, the baseline method (Shen et al., 2019) either changes image identity or confounds semantic properties, or both.

is easily extended to multiple attributes simultaneously. In addition, we incorporate a perceptual loss (Li et al., 2019) and an adversarial loss that help preserve image identity and photo-realism during manipulation.

We compare our method to several popular frameworks, from existing image-to-image translation methods (Choi et al., 2020; Wu et al., 2019; Zhu et al., 2017a) to latent space transformation-based approaches (Shen et al., 2019; Voynov & Babenko, 2020). We mention that prior work primarily uses qualitative evaluation. In contrast, we propose a quantitative evaluation to measure *controllability*. Both qualitative and quantitative results provide evidence that our approach outperforms prior work in terms of quality of the semantic image manipulation, while maintaining image identity.

## 2 Related work

**Generative Adversarial Networks (GANs)** (Goodfellow et al., 2014) have significantly improved realistic image generation in recent years (Brock et al., 2018; Jolicoeur-Martineau, 2019; Karras et al., 2017; 2019a;b; Park et al., 2019; Zhang et al., 2018). For this, a GAN formulates a 2-player non-cooperative game between two deep nets: (1) a generator that produces an image given a random noise vector sampled from a known prior distribution, usually a normal or a uniform distribution; (2) a discriminator whose input is synthetic and real data, which is to be differentiated.

**Semantic image editing** seeks to automate image manipulation of semantics. Encouraged by the success of deep nets, recent works have applied deep learning methods for semantic image editing tasks such as style transfer (Li et al., 2017; Luan et al., 2017), image-to-image translation (Choi et al., 2018; 2020; Isola et al., 2017; Lee et al., 2020; Wang et al., 2018; Wu et al., 2019; Zhu et al., 2017a) and discovering semantically meaningful directions in a GAN latent space (Härkönen et al., 2020; Jahanian et al., 2019; Plumerault et al., 2020; Shen et al., 2019; Voynov & Babenko, 2020). Note that our task is an extended version of semantic image editing that requires more comprehensive control to satisfy user-desired operations. Therefore, most aforementioned approaches do not meet the requirements. Nonetheless, we categorize the approaches that are most relevant to our task into two groups: (A) image editing via manipulation in image space, and (B) image editing via latent space navigation.

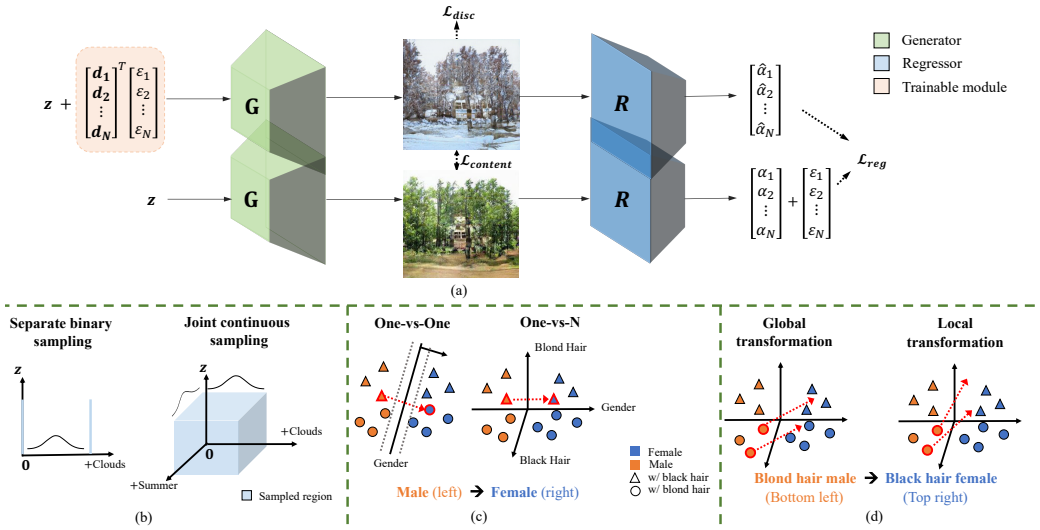


Figure 2: Overall framework (top row); 3 training strategies different from the prior work (bottom row). In (a), $G$ takes $\boldsymbol{z}$ and an edited latent vector separately to synthesize images. $\boldsymbol{T} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_N\}$ are the trainable latent-space directions and $\boldsymbol{\varepsilon}$ represents transformation extent. Original and edited image attributes, $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\alpha}}$, are predicted by the pre-trained regressor $R$. The discriminator loss $\mathcal{L}_{\text{disc}}$ (the discriminator $D$ not shown due to limited space), the regression loss $\mathcal{L}_{\text{reg}}$, and the perceptual loss $\mathcal{L}_{\text{content}}$ are used to update $\boldsymbol{T}$. We explain (b-d) in Sec. 3.2.

**Image-space editing using GANs** directly manipulates an image for targeted editing. Early work (Isola et al., 2017) used GANs to implement semantic translations between two image domains with paired data, e.g., day to night. Follow-up work focused on multi-modal (Bhattarai & Kim, 2020; Choi et al., 2018; 2020; Lee et al., 2020; Wang et al., 2018; Wu et al., 2019) and unpaired image domain translation (Zhu et al., 2017a). In this case, they primarily consider binary (on/off) attribute changes, regardless of whether the process is dynamic, for example, day to night.

**Latent-space editing via GANs** has received an increasing amount of recent interest. Most prior work focused on identifying semantically meaningful directions in the latent space of GANs, so that shifting latent vectors towards these directions achieves the desired image manipulation. Recent papers found semantics in the latent space of GANs, such as color transformations and camera movements (Jahanian et al., 2019), or face attribute changes (Shen et al., 2019), such as smile. Other work considered unsupervised methods (Härkönen et al., 2020; Voynov & Babenko, 2020) to discover interpretable latent space directions. However, additional challenges inherent to unsupervised manipulation of the latent space arise, and have not been addressed in prior work, e.g., direction quality with regard to degree control and image identity preservation.

# 3 Method

## 3.1 Problem Statement

We consider controllable semantic image editing via latent space navigation in GANs. We begin with a *fixed* GAN model that consists of a generator $G$ and a discriminator $D$. The input of $G$ is a latent vector $\boldsymbol{z} \in \mathbb{R}^m$ from a known distribution $\mathcal{Z}$. Here, $m$ is the dimensionality of the latent space. Given $N$ attributes, we aim to discover semantically meaningful latent-space GAN directions, $\boldsymbol{T} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N\}$, to manipulate the attributes of synthetic images $G(\boldsymbol{z})$ with an assigned step size $\boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_N\}$, where $\boldsymbol{d}_i \in \mathbb{R}^m$ for all $i \in \{1, \ldots, N\}$. In the end, each attribute of an edited image can be changed with the corresponding degree $\boldsymbol{\varepsilon}$ from $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_N\}$, the original attributes of $G(\boldsymbol{z})$ predicted by a regressor $R$.

## 3.2 Proposed Approach

We provide an overview of our method in Fig. 2 (a), and further illustrate how our approach differs from prior work in Fig. 2 (b-d).

**Overview.** As shown in Fig. 2 (a), we employ a GAN model consisting of a generator $G$ and a discriminator $D$ (not shown due to limited space), as well as a regressor $R$, all of which are pre-trained. Our goal is to find latent directions $\boldsymbol{T}$ which provide attribute specific image edits. At each training step, $G$ takes $\boldsymbol{z}$ and an edited latent vector referred to as $\boldsymbol{z}'$. We follow prior work which suggests that direction vectors in latent space of GANs permit image editing (Jahanian et al., 2019). Formally, given a transformation degree $\boldsymbol{\varepsilon}$, we obtain the latent-space edit as $\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{T}\boldsymbol{\varepsilon} = \boldsymbol{z} + \sum_{i=1}^{N} \varepsilon_i \boldsymbol{d}_i$. $G$ provides the recovered and edited images $G(\boldsymbol{z})$ and $G(\boldsymbol{z}')$, which are separately processed by the regressor $R$ to predict attribute values. The original attribute of $G(\boldsymbol{z})$ is $\boldsymbol{\alpha} = R(G(\boldsymbol{z}))$. The (pseudo-) ground truth and predicted attribute of $G(\boldsymbol{z}')$ are $\boldsymbol{\alpha}'$ and $\hat{\boldsymbol{\alpha}}'$, respectively, where $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \boldsymbol{\varepsilon}$ and $\hat{\boldsymbol{\alpha}}' = R(G(\boldsymbol{z}'))$.

The goal of $\boldsymbol{T}$ is to transform $\boldsymbol{z}$ by adding semantically meaningful information such that the corresponding output image $G(\boldsymbol{z}')$ exhibits attribute changes $\boldsymbol{\varepsilon}$ from $\boldsymbol{\alpha}$. In practice, we normalize the range of attribute values to $[\mathbf{0}, \mathbf{1}]$, i.e., both $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}' \in [\mathbf{0}, \mathbf{1}]$. During training, we maintain the unit range by controlling the given $\boldsymbol{\varepsilon}$. Formally, $\boldsymbol{\varepsilon}$ is drawn from a distribution $\mathcal{D}_\varepsilon$ uniform in $[-1, 1]^N$ while considering the constraint that $\mathbf{0} \leq \boldsymbol{\alpha} + \boldsymbol{\varepsilon} \leq \mathbf{1}$.

**Objective function.** To find $\boldsymbol{T}$ we minimize the weighted objective:

$$\min_{\boldsymbol{T}} \mathcal{L} = \lambda_1 \mathcal{L}_{\text{reg}} + \lambda_2 \mathcal{L}_{\text{disc}} + \lambda_3 \mathcal{L}_{\text{content}}. \tag{1}$$

The objective is only used for optimizing $\boldsymbol{T}$, while the other modules remain fixed. Hyperparameters $\lambda_1, \lambda_2$, and $\lambda_3$ adjust the loss terms.

The regression loss $\mathcal{L}_{\text{reg}}$ assesses whether $\boldsymbol{T}$ performs the transformations indicated by $\boldsymbol{\varepsilon}$. We express the regression loss via a binary cross entropy:

$$\mathcal{L}_{\text{reg}} = \quad \mathbb{E}_{\boldsymbol{z} \sim \mathcal{Z}, \boldsymbol{\varepsilon} \sim \mathcal{D}_\varepsilon} [-\hat{\boldsymbol{\alpha}}' \log \boldsymbol{\alpha}' - (\mathbf{1} - \hat{\boldsymbol{\alpha}}') \log (\mathbf{1} - \boldsymbol{\alpha}')]. \tag{2}$$

Note that $\boldsymbol{\alpha}'$ is from the distribution generated by $\boldsymbol{z}$ and $\boldsymbol{\varepsilon}$. Please refer to the appendix C.1 for more details about this distribution.

The second loss term $\mathcal{L}_{\text{disc}}$ is computed using the discriminator $D$ and measures the quality of the generated images, i.e.,

$$\mathcal{L}_{\text{disc}} = \quad \mathbb{E}_{\boldsymbol{z}' \sim \mathcal{Z}' | \boldsymbol{z}} \log \left[1 - D(G(\boldsymbol{z}'))\right]. \tag{3}$$

Here we refer to the distribution of $\boldsymbol{z}'$ via $\mathcal{Z}'$ conditioned on $\boldsymbol{z}$, formally written as $\mathcal{Z}'|\boldsymbol{z}$.

Lastly, we use a content loss $\mathcal{L}_{\text{content}}$, often also referred to as perceptual loss. It is designed to estimate the distance between two images and it is employed to maintain the image identity during the transformation. For this we use the content loss term

$$\mathcal{L}_{\text{content}} = \quad \mathbb{E}_{\boldsymbol{z} \sim \mathcal{Z}, \boldsymbol{z}' \sim \mathcal{Z}' | \boldsymbol{z}} \sum_{i \in \mathcal{D}_{\text{content}}} \|F_i(G(\boldsymbol{z}')) - F_i(G(\boldsymbol{z}))\|_2^2, \tag{4}$$

where $F_i(\cdot)$ is a feature function that extracts intermediate features of images. $\mathcal{D}_{\text{content}}$ indicates layers being used from a pre-trained model. We approximate the aforementioned expectations by empirical sampling. We defer algorithm details to the appendix A.

**Joint-distribution sampling and training.** The regressor operates in a multi-label setting, i.e., each data possesses multiple attributes. To ensure the disentanglement of attribute edits we sample synthetic images from the entire data distribution and find all transformations at once (illustrated in Fig. 2 (b,c) right). In contrast, Shen et al. (2019) prepare training samples on the two opposing data subsets with regards to an attribute, e.g., no clouds *vs*. many clouds, and find the directions with $N$ one-vs-one classifiers (sketched in Fig. 2 (b,c) left).

**Transformation module $\boldsymbol{T}$.** We study two types of transformations $\boldsymbol{T}$: i) *global* and ii) *local*. A *global* transformation $\boldsymbol{T}$ refers to a semantic latent-space transformation identical for all $\boldsymbol{z}$ during inference. This is illustrated via parallel red dashed arrows in Fig. 2 (d) left. The global directions are commonly used (Härkönen et al., 2020; Jahanian et al., 2019; Shen et al., 2019; Viazovetskyi et al., 2020). However, a globally identical direction might not serve all data. In contrast, a *local* transformation $\boldsymbol{T}$ is a function of $\boldsymbol{z}$ which provides various directions for different $\boldsymbol{z}$ (in Fig. 2 (d) right, shown via non-parallel red arrows). Formally, $\boldsymbol{d}_i = f_\theta^i(\boldsymbol{z})$, where $f_\theta^i$ is implemented via a deep net. In Sec. 4 we show that the local transformation $\boldsymbol{T}$ succeeds on failure cases of the global transformation.

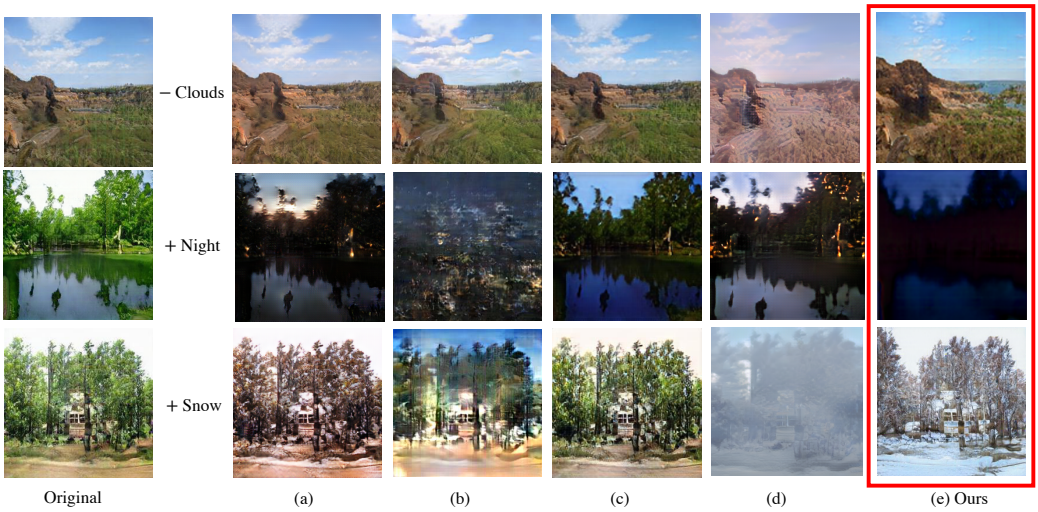


Figure 3: Comparison of image-space editing models with respect to "removing clouds" (top), "enhancing night" (middle) and "adding snow" (bottom). The original images (col.1) are followed by a given editing task. From (a-e): (a) CycleGAN (Zhu et al., 2017a), (b) StarGAN v2 (Choi et al., 2020), (c) RelGAN (Wu et al., 2019), (d) DRIT++ (Lee et al., 2020), and (e) Ours.

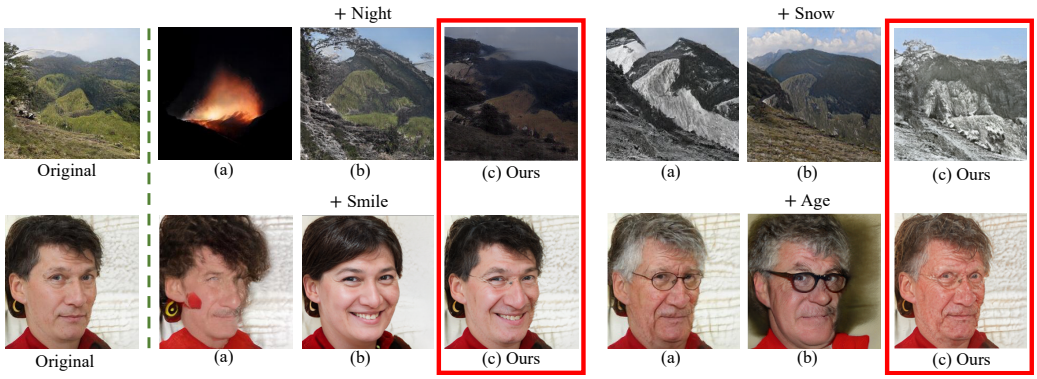


Figure 4: Comparison of latent-space editing approaches on StyleGAN2. (a) Shen et al. (2019); (b) Voynov & Babenko (2020); (c) Ours. Original synthetic image (col.1); edited results of a semantic manipulation noted on top (col.2-4; col.5-7). It is mentioned in Shen & Zhou (2020) that "age", "gender", and "glasses" directions are hard to disentangle potentially due to data bias, while our results suggest a better direction disentanglement.

## 4 Experiments

We introduce datasets, implementation details, evaluation metrics, and show results next.

**Datasets.** We evaluate the proposed approach on two types of datasets: (i) face datasets – FFHQ (Karras et al., 2019a), CelebA (Liu et al., 2018) and CelebA-HQ (Karras et al., 2017), commonly used in prior work (Härkönen et al., 2020; Karras et al., 2017; 2019a;b; Shen et al., 2019; Viazovetskyi et al., 2020). (ii) natural scene datasets – Transient Attribute Database (Laffont et al., 2014) and MIT Places2 data (Zhou et al., 2017), which contain attributes suitable for continuous semantic image editing. We briefly introduce the scene datasets:

- *Transient Attribute Database* (Laffont et al., 2014): It contains 8,571 scene images with 40 attributes in 5 categories including lighting (e.g., "bright"), weather (e.g., "cloudy") , seasons (e.g., "winter"), subjective impressions (e.g., "beautiful"), and additional attributes (e.g., "dirty"). Each attribute is annotated with a real-valued score between 0 and 1, where 0 indicates absence of the attribute.

***Question1***: *Compared to the picture on the left, how cloudy is the picture on the right?*

***Question2***: *Ignoring the attribute clouds, determine if the two pictures record the same scene.*

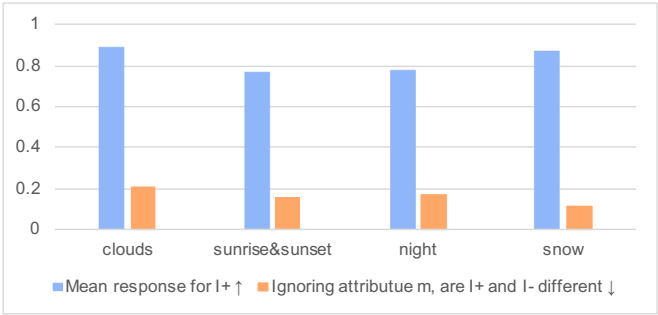


Figure 5: Evaluation of image editing and image identity preservation via a user study. Blue bars show the predicted presence of a given attribute for images ($I+$, higher is better). The orange bars measure the difference between paired images (lower is better).

- *MIT Places2 data* (Zhou et al., 2017): With the provided category annotations, we select the natural outdoor scenes, obtaining a total of 144,543 images.

**Implementation details.** We choose $\lambda_1 = 10, \lambda_2 = 0.05, \lambda_3 = 0.05$ in Eq. (1) and compute the perceptual loss using the $\text{conv1_2}, \text{conv2_2}, \text{conv3_2}, \text{conv4_2}$ activations in a VGG-19 network (Simonyan & Zisserman, 2014) pre-trained on the ImageNet dataset (Russakovsky et al., 2015). We train $\boldsymbol{T}$ for 50k iterations with a batch size of 4. An Adam optimizer is used with a learning rate of $1e-4$. We evaluate the proposed method on the $\mathcal{W}$ space of StyleGAN2 and the $\mathcal{Z}$ space of PGGAN. For the attribute regressor $R$, we adopt a ResNet-50 (He et al., 2016) for attribute prediction on the Transient Attribute (Laffont et al., 2014) and the CelebA (Liu et al., 2018) data. The last fully connected layer in the ResNet-50 is replaced by a linear layer with an output dimension of 40. We train the regressors for 500 epochs and use the weights with the best validation mean squared error (MSE) on CelebA and the best test MSE on the Transient Scene Database. The GAN follows the StyleGAN2 architecture and is pretrained with 200k iterations on a union of the two natural scene datasets using a resolution of $256 \times 256$. The training batch size is 32 and an Adam optimizer is used with a learning rate of $2e-3$. For the face datasets, we use FFHQ pre-trained weights[1] of StyleGAN2, and CelebA-HQ weights of PGGAN[2].

**Baselines.** We compare our approach to several popular image-to-image translation approaches (Choi et al., 2020; Wu et al., 2019; Zhu et al., 2017a) and latent space direction discovery methods (Shen et al., 2019; Voynov & Babenko, 2020). Since Choi et al. (2020); Shen et al. (2019); Zhu et al. (2017a) cannot deal with continuous attributes, we split the data into 2 domains using a threshold value of $0.5$ for each of the 40 attributes. Afterward, the models are trained on 2 contrast domains, e.g., with or without a certain attribute. We use the official code of all baseline methods and rigorously follow their training steps. Note that work by Voynov & Babenko (2020) is unsupervised, i.e., the latent-space directions are human interpreted. To avoid bias during the selection of directions, we use the attribute regressor to automatically identify the most significant directions that can edit predetermined attributes. The details of preparing the baselines are in the appendix C.2.

**Evaluation metrics.** There is no good numerical metric to evaluate image editing (Shen et al., 2019; Voynov & Babenko, 2020). In an attempt to address this concern, we automate quantitative evaluation based on a property that editing of attributes should maintain image identity. To achieve this, we employ a popular image identity recognition model[3] pre-trained on the VGGface2 data (Cao et al., 2018). Cosine similarity is used to represent the similarity between paired original and edited images (Cao et al., 2018). In addition, we evaluate changes of the other independent attributes using the pre-trained regressor.

### 4.1 Results on natural scene datasets

**Comparison to image-to-image translation:** As shown in Fig. 3 (a-d), we observe image-to-image translation to perform poorly when editing image details (e.g., removing clouds) and sometimes

[1]https://github.com/rosinality/stylegan2-pytorch
[2]https://pytorch.org/hub
[3]https://github.com/ox-vgg/vgg_face2

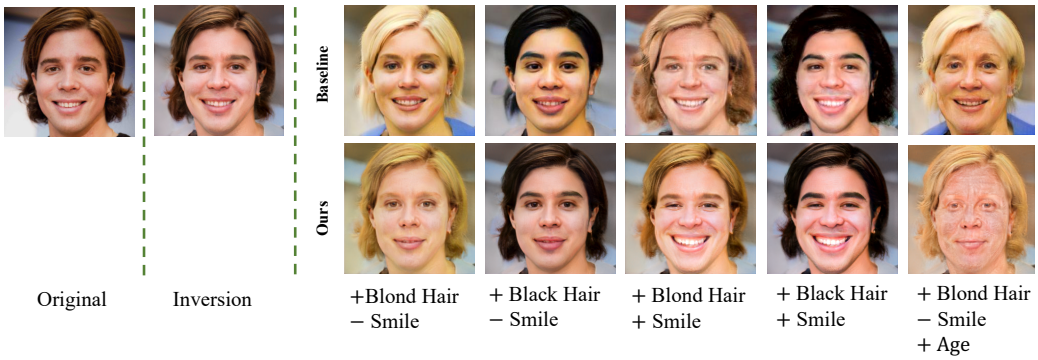


Figure 6: Real image synthesis with multiple attributes. Shen et al. (2019) (top row); Ours (bottom row): original real image (col.1); inverted result (col.2) from Abdal et al. (2019); edited results (col.3-7) with target attribute changes shown at the bottom.

contains artifacts (orange spots in (a) and (b)). In contrast, our model shows overall compelling transfer performance on all these attributes. We show more results in the appendix D.2.

**Comparison to latent-space translation:** Inspecting Fig. 1, we observe our approach to improve upon work by Shen et al. (2019) in attribute edit quality and number of artifacts. Similarly, in Fig. 4, we notice that our method performs well in attribute editing, and has a better ability to preserve image identity when adding "night" and "snow." We show more results in the appendix D.

**Quantitative evaluation:** We follow a user study proposed by Kowalski et al. (2020) to evaluate controllability of our scene edits. Concretely, given an image $I = G(\boldsymbol{z})$, we generate paired images $(I+, I-)$ that have opposing values for attribute $n$. 40 users took the test to answer: (a) the presence and strength of the attribute $n$ in $I+$, and (b) if $I+$ and $I-$ are identical when ignoring the attribute. Both types of questions have options on a 5-level scale from 0 ("not at all") to 1 ("totally"). Ideally, the response of question (a) should be 1 and (b) is 0. We test performance of image editing with "clouds", "sunrise&sunset", "night" and "snow" attributes. A question example and statistical results over 50 image pairs are shown in Fig. 5. Statistical evidence supports our claim that the proposed approach performs well with regards to image edits while maintaining image identity.

| | Smile | | | Hair color | | | Smile + Hair color | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lvert\hat{\varepsilon}\rvert$ | (0, .3] | (.3, .6] | (.6, .9] | (0, .3] | (.3, .6] | (.6, .9] | (0, .3] | (.3, .6] | (.6, .9] |
| Shen et al. | .202<br>5$e$-2 | .204<br>6$e$-2 | .224<br>4$e$-2 | .256<br>1$e$-2 | .272<br>2$e$-2 | .277<br>2$e$-2 | .299<br>5$e$-3 | .318<br>1$e$-2 | .329<br>3$e$-2 |
| Voynov et al. | .115<br>6$e$-3 | .211<br>4$e$-2 | .277<br>7$e$-3 | .162<br>1$e$-2 | .166<br>3$e$-2 | .177<br>9$e$-3 | .155<br>4$e$-3 | .220<br>4$e$-2 | .284<br>2$e$-2 |
| Ours | **.085**<br>4$e$-2 | **.084**<br>3$e$-3 | **.098**<br>4$e$-3 | **.075**<br>7$e$-4 | **.083**<br>5$e$-3 | **.084**<br>5$e$-3 | **.088**<br>7$e$-3 | **.111**<br>4$e$-3 | **.134**<br>4$e$-3 |

Table 1: Quantitative evaluation of numerical changes on the other semantically independent attributes (lower is better). (*first row*) edited attributes; (*second row*) the absolute changing range of the edited attributes; (*bottom three rows*) averages (up) and standard deviations (down) in each row.

| | Smile | | | Hair color | | | Smile + Hair color | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lvert\hat{\varepsilon}\rvert$ | (0, .3] | (.3, .6] | (.6, .9] | (0, .3] | (.3, .6] | (.6, .9] | (0, .3] | (.3, .6] | (.6, .9] |
| Shen et al. | .918<br>4$e$-3 | .916<br>1$e$-3 | .907<br>5$e$-3 | .887<br>3$e$-2 | .877<br>4$e$-2 | .874<br>4$e$-2 | .877<br>6$e$-3 | .819<br>3$e$-2 | .801<br>4$e$-2 |
| Voynov et al. | .979<br>2$e$-3 | .896<br>4$e$-3 | .869<br>9$e$-3 | .955<br>6$e$-2 | .909<br>4$e$-2 | .904<br>3$e$-2 | .940<br>9$e$-3 | .829<br>4$e$-2 | .811<br>3$e$-2 |
| Ours | **.995**<br>7$e$-4 | **.994**<br>1$e$-3 | **.992**<br>9$e$-4 | **.993**<br>5$e$-4 | **.993**<br>1$e$-4 | **.993**<br>2$e$-4 | **.992**<br>9$e$-4 | **.986**<br>2$e$-3 | **.984**<br>4$e$-4 |

Table 2: Quantitative evaluation on image identity preservation (higher is better). Notation is identical to Tab. 1.

(a) + Black hair (b) + Smile

Figure 7: Continuous editing results on PGGAN (Karras et al., 2017) with the "black hair" (left) and the "smile" (right) attribute.

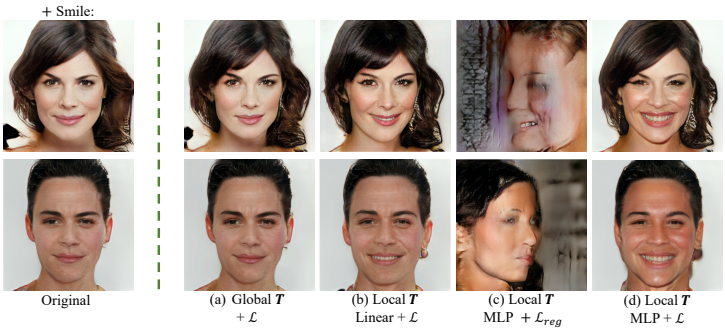


Figure 8: Comparison of global and local transformations on PGGAN. Original synthetic images (col.1); the edited results on enhancing "smile" (col.2-5): (a) a global $\boldsymbol{T}$, (b) a local $\boldsymbol{T}$, with linear-layer direction functions, (c) a local $\boldsymbol{T}$ with MLP direction functions trained via $\mathcal{L}_{\text{reg}}$ loss only, and (d) a local $\boldsymbol{T}$ with MLP direction functions training via the entire loss $\mathcal{L}$.

### 4.2 Results on face datasets

**Manipulation results on StyleGAN2:** The comparison on real images shown in Fig. 1 suggests that our method works well on the shown attribute edits. Further, the synthetic image edit results in Fig. 4 indicate that our edits are disentangled, while the baselines unexpectedly add "glasses" when aging the face. To increase the task difficulty, we edit real images with multiple attribute changes simultaneously. Results are summarized in Fig. 6, which highlights the controllability of our edits.

**Quantitative evaluation:** We measure the changing degrees of the other independent attributes and image identity when editing attributes with various degrees $\hat{\varepsilon}$. Here $\hat{\varepsilon}$ is a predicted changing degree on a target attribute by $R$. For a thorough comparison, we evaluate the performance on 3 segments according to the absolute value of $\hat{\varepsilon}$, i.e., $|\hat{\varepsilon}|$ in the range of $(0, 0.3]$, $(0.3, 0.6]$ and $(0.6, 0.9]$. Evaluation on "smile," "hair color," and "smile+hair color" attributes are shown in Tab. 1 and Tab. 2. The "hair color" attribute includes "blond" and "black" colors where we average the results on both cases. For multiple attribute editing, "smile + hair color," we evaluate the case that both the two targeted attributes change within the $|\hat{\varepsilon}|$ range. We use around 1k original images, generate 10k edited images with regard to each target attribute, and repeat the experiment 3 times. The averaged results and the standard deviations presented in Tab. 1 and Tab. 2 suggest that our model outperforms the baselines with regard to disentanglement and image identity preservation.

**Manipulation results on PGGAN:** For PGGAN, we use local transformations parameterized by three-layer perceptrons with leaky ReLU activations. We show the detailed MLP architecture in the appendix B.1. Editing results in Fig. 7 suggest that the proposed approach is able to continuously edit faces with various semantic attributes. More results are shown in the appendix. We also study the differences between local and global directions in Fig. 8. Particularly, a global $\boldsymbol{T}$ (in Fig. 8 (a)) is a matrix where each column $i$ is given by $\boldsymbol{d}_i$. For a local $\boldsymbol{T}$ (in Fig. 8 (b)) $\boldsymbol{d}_i$ is given by a fully-connected layer with identical input and output dimensions. Fig. 8 (c) and (d) use the MLP components with the aforementioned structure. We use $\mathcal{L}_{\text{reg}}$ to train $\boldsymbol{T}$ in (c), while the total loss $\mathcal{L}$ is used for (d). The results in Fig. 8 indicate that a local $\boldsymbol{T}$, with either linear or MLP structure works well for face attribute edits. Moreover, our loss $\mathcal{L}$ stabilizes the training process of $\boldsymbol{T}$.

## 5 Conclusion

We propose an effective approach to semantically edit images by transferring latent vectors towards meaningful latent space directions. The proposed method enables continuous image manipulations with respect to various attributes. Extensive experiments highlight that our model achieves state-of-the-art performance for targeted image manipulation.

## REFERENCES


Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

Binod Bhattarai and Tae-Kyun Kim. Inducing optimal attribute representations for conditional gans. *ECCV*, 2020.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv*, 2018.

Q. Cao, L. Shen, W. Xie, et al. Vggface2: A dataset for recognising faces across pose and age. In *IEEE FG*, 2018.

Yunjey Choi, Minje Choi, Munyoung Kim, et al. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, et al. Stargan v2: Diverse image synthesis for multiple domains. *CVPR*, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial nets. In *NeurIPS*, 2014.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, et al. Ganspace: Discovering interpretable gan controls. *arXiv*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, et al. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. *ICLR*, 2019.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *ICLR*, 2019.

Tero Karras, Timo Aila, Samuli Laine, et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019a.

Tero Karras, Samuli Laine, Miika Aittala, et al. Analyzing and improving the image quality of stylegan. *arXiv*, 2019b.

Marek Kowalski, Stephan J Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. *ICML*, 2020.

Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, et al. Transient attributes for high-level understanding and editing of outdoor scenes. *TOG*, 2014.

H.Y. Lee, H.Y. Tseng, Q. Mao, et al. Drit++: Diverse image-to-image translation via disentangled representations. *IJCV*, 2020.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Others. Controllable text-to-image generation. In *NeurIPS*, 2019.

Y. Li, C. Fang, J. Yang, et al. Universal style transfer via feature transforms. In *NeurIPS*, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, et al. Large-scale celebfaces attributes (celeba) dataset. 2018.

F. Luan, S. Paris, and E. andothers Shechtman. Deep photo style transfer. In *PCVPR*, 2017.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *ICLR*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, et al. Imagenet large scale visual recognition challenge. *ICCV*, 2015.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv*, 2020.

Yujun Shen, Jinjin Gu, Xiaoou Tang, et al. Interpreting the latent space of gans for semantic face editing. 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. Stylegan2 distillation for feed-forward image manipulation. *arXiv*, 2020.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. *ICML*, 2020.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, et al. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, et al. Relgan: Multi-domain image-to-image translation via relative attributes. In *ICCV*, 2019.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, et al. Self-attention generative adversarial networks. *PMLR*, 2018.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, et al. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, et al. Toward multimodal image-to-image translation. In *NeurIPS*, 2017b.

# APPENDIX

## A. ALGORITHM

The overall procedure of the proposed method in Algorithm 1, taking local transformation $\boldsymbol{T}_\theta$ as an example.

**Algorithm 1** Training Procedure

**Input:** A pre-trained GAN with $G$, $D$, and input noise distribution $\boldsymbol{z} \sim \mathcal{Z}$; a pre-trained regressor $R$ for predictions on N attributes; an initialized $\boldsymbol{T}_\theta$; a learning rate $\mu_\theta$ for $\boldsymbol{T}_\theta$; max iteration number $M$.

**for** iteration $m = 1, \ldots, M$ **do**
  Sample random noise $\boldsymbol{z} \sim \mathcal{Z}$, and $\boldsymbol{\varepsilon}$
  Compute internal attributes for a synthetic image, i.e., $\boldsymbol{\alpha} = R(G(\boldsymbol{z}))$
  Compute the actual shift value $\boldsymbol{\delta} = CLIP(\boldsymbol{\alpha} + \boldsymbol{\varepsilon}, (0, 1)) - \boldsymbol{\alpha}$
  Compute the transformed latent vector $\boldsymbol{z}' = \boldsymbol{z} + \boldsymbol{T}_\theta(\boldsymbol{z})\boldsymbol{\delta}$
  Compute $I' = G(\boldsymbol{z}')$, $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \boldsymbol{\delta}$
  Compute attribute predictions $\hat{\boldsymbol{\alpha}}' = R(I')$
  Compute the loss $\mathcal{L}$
  Update $\theta \leftarrow \theta - \mu_\theta * \frac{\partial \mathcal{L}}{\partial \theta}$
**end for**

**Return:** $\boldsymbol{T}_\theta$

## B. METHOD DETAILS

### B.1. METHOD ON PGGAN

We apply a multi-layer perceptron (MLP) to parameterize a latent space path on PGGAN. Fig. 9 shows the MLP architecture that we use.

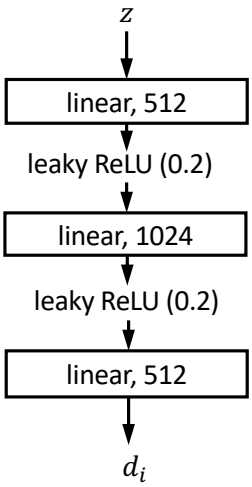


Figure 9: Example MLP architecture on PGGAN.

In practice, we find that normalizing the direction vector $\boldsymbol{d}_i$ helps preserve image identity during manipulation on PGGAN. Formally, the transformed latent vector can be written as

$$\boldsymbol{z}' = \boldsymbol{z} + \lambda \boldsymbol{\varepsilon} \frac{\boldsymbol{d}_i}{\|\boldsymbol{d}_i\|}, \tag{5}$$

where $\lambda$ is a weight adjusting the direction scale. We set $\lambda = 3$.

# C. EXPERIMENTAL DETAILS

## C.1 THE EXPECTATION IN $\mathcal{L}_{\text{REG}}$

We define the regression loss $\mathcal{L}_{\text{reg}}$ as

$$\mathcal{L}_{\text{reg}} = \quad \mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z},\boldsymbol{\varepsilon}\sim\mathcal{D}_{\varepsilon}}[-\hat{\boldsymbol{\alpha}}' \log \boldsymbol{\alpha}' - (\mathbf{1} - \hat{\boldsymbol{\alpha}}') \log (\mathbf{1} - \boldsymbol{\alpha}')], \tag{6}$$

where $\boldsymbol{\alpha}'$ is from the distribution generated by $\boldsymbol{z}$ and $\boldsymbol{\varepsilon}$. We now discuss how we derive the distribution of $\boldsymbol{\alpha}'$. To begin, $\boldsymbol{z}$ is sampled from $\mathcal{Z}$ and $\boldsymbol{\varepsilon}$ is sampled from $\mathcal{D}_{\varepsilon}$. The other variables are generated by $\boldsymbol{z}$ and $\boldsymbol{\varepsilon}$:

$$\begin{aligned} \boldsymbol{z}' &= \boldsymbol{z} + \boldsymbol{T}\boldsymbol{\varepsilon} \\ \boldsymbol{\alpha}' &= R(G(\boldsymbol{z})) + \boldsymbol{\varepsilon} \\ \hat{\boldsymbol{\alpha}}' &= R(G(\boldsymbol{z}')). \end{aligned}$$

As a result, the distribution of $\boldsymbol{\alpha}'$ is dependent on $\mathcal{Z}$ and $\mathcal{D}_{\varepsilon}$.

## C.2 UNSUPERVISED LATENT-SPACE EDITS OF GANS

Voynov & Babenko (2020) learn a matrix where each column is a direction. Yet, Voynov & Babenko (2020) require a human to interpret the learnt directions. To avoid bias during the selection of directions in comparison, we use the pre-trained attribute regressor to automatically identify the most significant directions. Concretely, we first generate 100 images and edit them with target attributes of various degrees. Next, we use the regressor $R$ on the edited images to predict attributes. We choose for an attribute edit the direction on which all the edited images have the overall highest changing response on the target attribute.

# D. ADDITIONAL RESULTS

## D.1 SCENE IMAGE EDITS ON STYLEGAN2

We show more results of our method on continuous scene image edits in Fig. 10 - 13. The semantic edit attributes include "clouds" (Fig. 10), "brightness" (Fig. 11), "snow" (Fig. 12), and "summer" (Fig. 13).

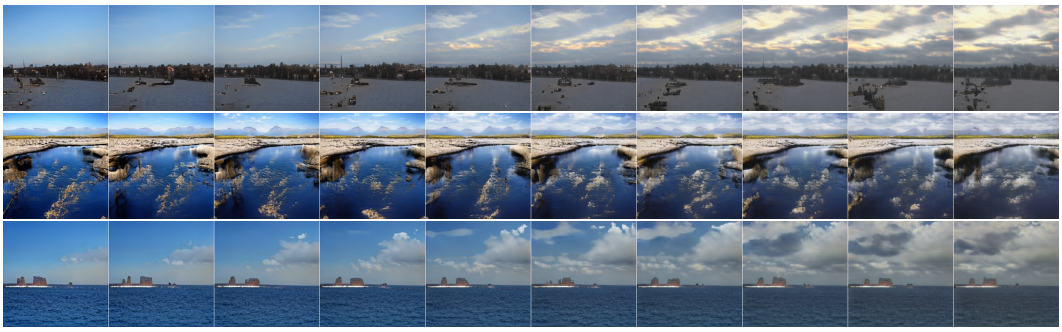

Figure 10: Continuous image edits using StyleGAN2 (Karras et al., 2019b) on the "clouds" attribute.

## D.2 SCENE IMAGE EDITS COMPARED WITH IMAGE-TO-IMAGE TRANSLATION WORK

We show more results for image-space editing methods in Fig. 14.

## D.3 SCENE IMAGE EDITS WITH MULTIPLE ATTRIBUTES

Editing of scene images with multiple attributes simultaneously is shown in Fig. 15.

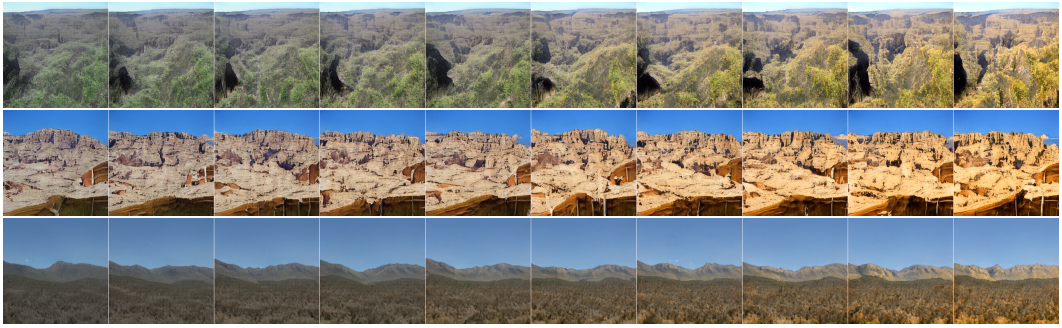

Figure 11: Continuous image edits using StyleGAN2 (Karras et al., 2019b) on the "brightness" attribute.

Figure 12: Continuous image edits using StyleGAN2 (Karras et al., 2019b) on the "snow" attribute.

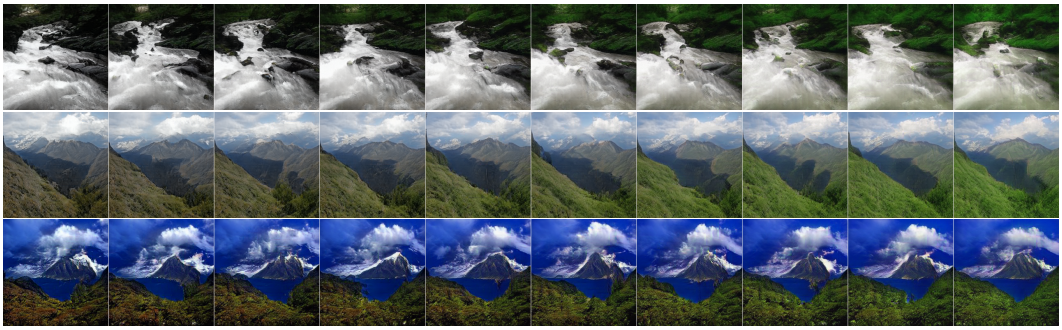

Figure 13: Continuous image edits using StyleGAN2 (Karras et al., 2019b) on the "summer" attribute.

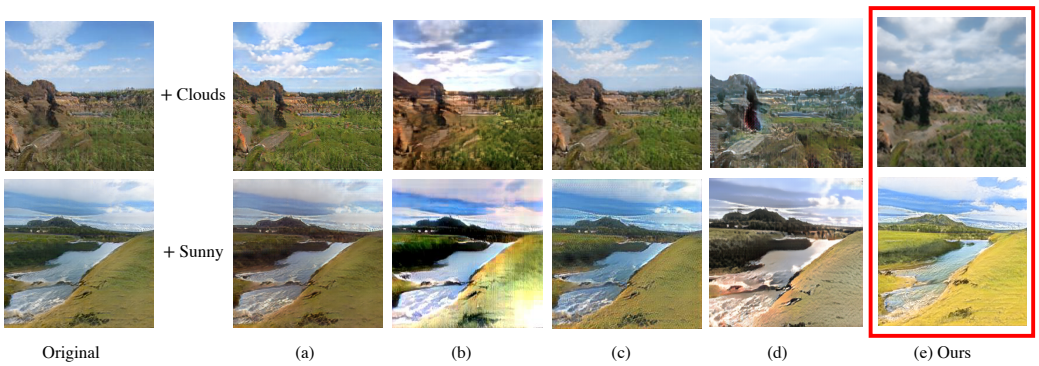


Figure 14: Comparison of image-space editing models with respect to "adding clouds" (top), and "enhancing sunny" (middle) attributes. The original images (col.1) are followed by a given editing task. From (a-e): (a) CycleGAN (Zhu et al., 2017a), (b) StarGAN v2 (Choi et al., 2020), (c) RelGAN (Wu et al., 2019), (d) DRIT++ (Lee et al., 2020), and (e) Ours.

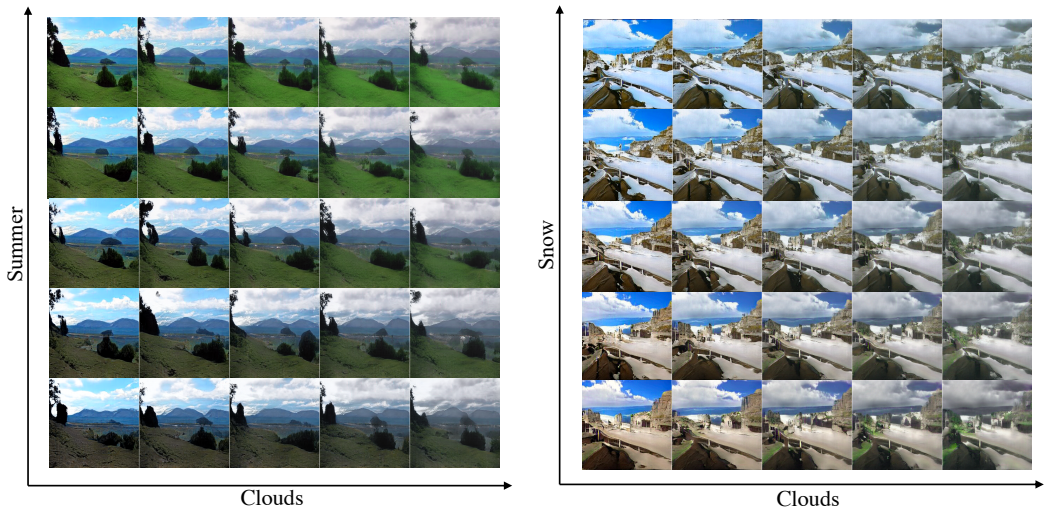


Figure 15: Scene edits with multiple attributes. The left figure shows the manipulation for a given image towards “summer” and “clouds” attributes simultaneously. The right figure shows the manipulation towards “snow” and “clouds”.

Figure 16: Continuous image edits using StyleGAN2 (Karras et al., 2019b) on the “gender” attribute.

### D.4 FACE IMAGE EDITS

Editing of face images are shown in Fig. 16 -Fig. 17. The semantic edit attributes includes “gender” (Fig. 16), and “smile” (Fig. 17).

Figure 17: Continuous image edits using PGGAN (Karras et al., 2017) on the “smile” attribute.