CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing

ANONYMOUS AUTHOR(S)
SUBMISSION ID: 557

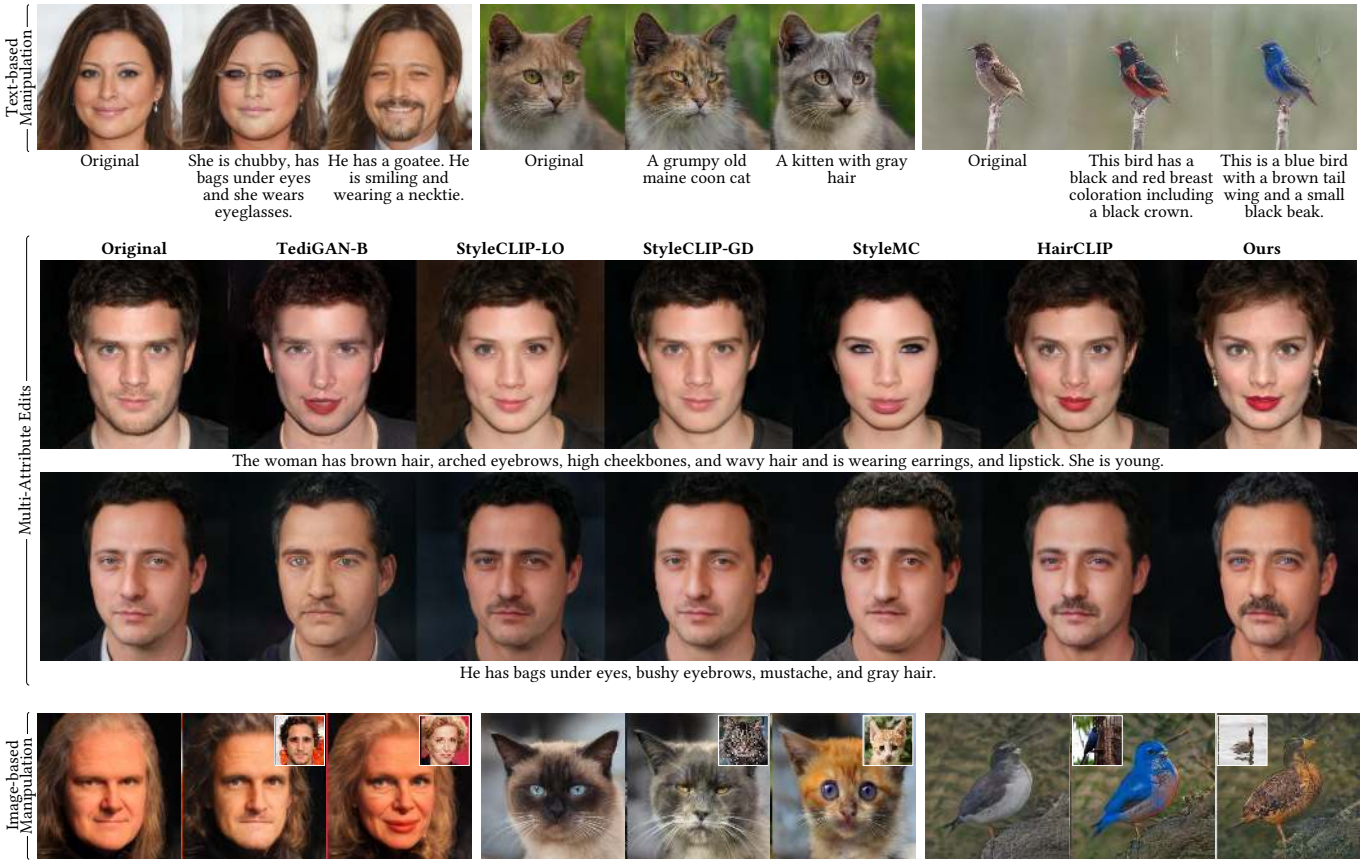


Fig. 1. **Multi-attribute real image manipulation with CLIPInverter.** We present *CLIPInverter* that enables users to easily perform semantic changes on images using free natural text. Our approach is not specific to a certain category of images and can be applied to many different domain (e.g., human faces, cats, birds) where a pretrained StyleGAN generator exists (*top*). Our approach specifically gives more accurate results for multi-attribute edits as compared to the prior work (*middle*). Moreover, as we utilize CLIP's semantic embedding space, it can also perform manipulations based on reference images without any training or finetuning (*bottom*).

Researchers have recently begun exploring the use of StyleGAN-based models for real image editing. One particularly interesting application is using natural language descriptions to guide the editing process. Existing approaches for editing images using language either resort to instance-level latent code optimization or map predefined text prompts to some editing directions in the latent space. However, these approaches have inherent limitations. The former is not very efficient, while the latter often struggles to effectively handle multi-attribute changes. To address these weaknesses, we present CLIPInverter, a new text-driven image editing approach that is able to efficiently and reliably perform multi-attribute changes. The core of our method is the use of novel, light-weight text-conditioned adapter layers integrated into pretrained GAN-inversion networks. We demonstrate that by conditioning the initial inversion step on the CLIP embedding of the target description, we are able to obtain more successful edit directions. Additionally, we use a CLIP-guided refinement step to make corrections in the resulting residual latent codes, which further improves the alignment with the text prompt. Our method outperforms competing approaches in terms of manipulation accuracy and photo-realism on various domains including human faces, cats, and birds, as shown by our qualitative and quantitative results.

0730-0301/2022/12-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS Concepts: • **Computing methodologies** → **Image manipulation**; **Neural networks**.

Additional Key Words and Phrases: Generative Adversarial Networks, Image-to-Image Translation, Image Editing

ACM Reference Format:
Anonymous Author(s). 2022. CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing. *ACM Trans. Graph.* 1, 1 (December 2022), 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The quality of images synthesized by Generative Adversarial Networks [Goodfellow et al. 2014] have reached a remarkable level in less than a decade. StyleGAN and its variants [Karras et al. 2021, 2019, 2020] are now capable of generating highly realistic images, while allowing control over the generation process by means of style mixing. Recent works [Härkönen et al. 2020; Shen et al. 2020] have demonstrated that StyleGAN learns disentangled attributes, making it possible to find directions in its latent space to generate images that possess such desired attributes. Consequently, there has been a growing interest in utilizing semantic editing directions in the latent space mostly for preset directions such as gender, face orientation, hair color.

Concurrent to the advances in generative modeling, we are also witnessing exciting breakthroughs in multimodal learning. For example, the recently proposed Contrastive Language-Image Pre-training (CLIP) model [Radford et al. 2021] provides an effective common embedding for images and text captions. Such an embedding, when combined with powerful GANs paves the road towards text-guided image editing, one of the most natural and intuitive ways of manipulating images. Hence, it comes with no surprise that several recent works [Kocasari et al. 2021; Li et al. 2020; Patashnik et al. 2021; Wei et al. 2022; Xia et al. 2021a] have focused on mapping target textual descriptions to editing directions in the latent space of StyleGAN. While some methods perform optimization in the latent space guided by CLIP [Patashnik et al. 2021; Xia et al. 2021a] or find global directions that can be applied to any input image [Kocasari et al. 2021; Patashnik et al. 2021], others train a separate mapper network for each type of textual edit [Patashnik et al. 2021] or a general mapper conditioned on reference images & textual descriptions [Wei et al. 2022]. None of these GAN based methods consider language-conditioning on the initial inversion phase. Finally, recent diffusion-based methods perform image manipulation either by fine-tuning pre-trained text-to-image models [Kawar et al. 2022; Valevski et al. 2022] or guiding the reverse diffusion by the CLIP model [Kim et al. 2022].

Directly optimizing a latent code based on CLIP similarity or finding global editing directions require instance-based optimization, causing long inference times. Training mappers for a single text prompt reduces the inference time to a single forward pass, but comes with the price of training time as separate mappers need to be trained for each text prompt. Moreover, these mappers that operate in the latent space do not directly consider the features of the original image as they take inverted latent codes as inputs from pretrained GAN inversion networks. In this study, we present a new approach, which we call *CLIPInverter*, to automatically edit an input image based on a target textual description by adjoining light-weight adapter modules to pretrained unconditional inversion methods. CLIPInverter includes a novel CLIP-conditioned adapter module (*CLIPAdapter*) that is attached to the pretrained encoder model to map both the input image and the target textual description to a residual latent code by utilizing the common CLIP embedding space. The residual latent code is then combined with the latent code of the input image obtained by the unconditional branch of the encoder, and is fed to a CLIP-guided correction module (*CLIPRemapper*) that applies a final correction by blending the latent codes with latent codes predicted from the CLIP embedding of the target textual description based on learnable blending coefficients. The final latent code is decoded by a pretrained and frozen StyleGAN2 generator to synthesize the manipulated image that reflects the desired changes while preserving the identity of the original subject as much as possible. Our encoder-adapters are lightweight networks that directly modulate image feature maps using text embeddings and they could be appended to many pretrained encoders. Our CLIP-guided correction module utilizes the CLIP text embeddings to enhance the manipulations of the generated images while preserving the photo-realism. Our method does not require any additional optimization on the latents and it successfully aplies manipulations using various text prompts in a single forward pass. Since we directly modulate feature maps extracted during the inversion phase, our method is capable of editing images much better than the competing approaches, especially in cases when there are multiple attributes present in the target textual description, as proven by our experiments. See Fig. 2 for an overview of our framework.

Our method finds a good balance for the distortion-editability tradeoff discussed by Tov et al. [2021]. Namely, our text-guided CLIPAdapter is employed to find an editing direction that is aligned with the given target description, specific to the input image. Since we are inverting to the $\mathcal{W}+$ space, we are able to achieve quite low distortion, as the identity in the input image is preserved in the manipulated image. However, we observe that not all of the attributes described in the target caption are captured in the manipulated image. We introduce the text-guided refinement module, CLIPRemapper to apply a final correction to the latent code. Essentially, CLIPRemapper finds a more editable region in the vicinity of the latent code we obtain from the previous stage. This process boosts the manipulation performance of our model massively, while keeping the distortion at a comparable level, as shown in our ablation study.

The main contribution of our work is to map the given target textual description containing multiple attributes to a residual latent code using the feed-forward encoder without requiring any optimization strategy. We demonstrate editing results for challenging cases where there are many attributes present in the target description. Our method is not restricted to a particular domain like commonly studied human faces, and we also evaluate our approach on birds and cats images. Exploring the multimodal nature of CLIP, instead of target textual descriptions, we can additionally use images or target textual descriptions containing vocabulary never seen during training as the guiding signal. Finally, we show that linearly interpolating between the original latent code and the updated latent code results in smooth image manipulations, providing a means for user to have control over the manipulation process.

We evaluate our method on a diverse set of datasets and provide detailed qualitative results and comparisons against the state-of-the-art models. Quantitative comparisons in language-guided editing still remains a challenge, as one needs to evaluate the manipulations from different aspects, such as accuracy, preservation of text-irrelevant details, photorealism etc. Current metrics are not suitable for evaluation as they do not consider some of these aspects at all. We propose two new metrics, Attribute Manipulation Accuracy (AMA), CLIP Manipulative Precision (CMP) to measure how accurately the manipulations are applied, and how well the text-irrelevant details are preserved. We perform quantitative comparisons against state-of-the-art models using these metrics along with FID. These comparisons as well as a user study that we conducted to evaluate perceptual realism and manipulation accuracy demonstrate the superiority of our approach over the prior work.

2 RELATED WORK

2.1 State-of-the-Art in GANs

In less than a decade, generative adversarial networks (GANs) [Goodfellow et al. 2014] have shown great progress in synthesizing realistic images. These advances are made possible by applying novel combinations of architectural changes and training strategies. For example, PGGAN [Karras et al. 2018] utilizes progressive growth of the generator to add increasingly new details to the synthesized images. The authors employ the Wasserstein Loss [Arjovsky et al. 2017] for more stable and robust training. BigGAN [Brock et al. 2019] uses residual connections and class labels in a large-scale model and introduce a truncation trick to synthesize high-quality images. The projection discriminator [Miyato and Koyama 2018] used in BigGAN allows it to model class-conditional distributions. StyleGAN [Karras et al. 2019] first maps a latent code to an intermediate latent space using a latent mapper. These intermediate latent codes determine the parameters of the AdaIN [Huang and Belongie 2017] layers introduced in the generator, to control the style of the generated image. StyleGAN2 [Karras et al. 2020] proposes several modifications in the layers of StyleGAN to improve image quality. More recently, StyleGAN3 [Karras et al. 2021] makes the underlying architecture fully equivariant to translation and rotation, further reducing the visible artifacts. StyleSwin [Zhang et al. 2021] utilizes transformers instead of convolutional neural networks to synthesize high quality images and GANformer [Hudson and Zitnick 2021] uses a bipartite structure to generate images with multiple objects. Among these alternatives, in our study, we utilize a pretrained StyleGAN2 model for a fair comparison with competing methods.

2.2 GAN Inversion

The aim of GAN Inversion [Zhu et al. 2016] is to map a given image back into the latent space of a pretrained GAN model. The generator then can be used as a decoder to reconstruct the image from the inverted latent code. This process enables manipulation on real images since the inverted latent codes can be modified in a way to reflect desired editing tasks. Below we discuss some representative works to highlight three main approaches to accomplish GAN Inversion – please refer to the recent survey [Xia et al. 2021b] for an in-depth discussion of various other inversion methods.

The optimization-based methods directly optimize a latent code that reconstructs the target image as close as possible using gradient descent [Abdal et al. 2019, 2020; Creswell and Bharath 2016; Tewari et al. 2020b]. This line of works is instance specific, and does not require any trainable modules. The learning-based methods invert an image by a learned encoder. This approach is similar to an autoencoder pipeline, where the pretrained generator acts as the decoder. Unconditional encoders [Alaluf et al. 2021b; Bai et al. 2022; Bau et al. 2019a; Richardson et al. 2021; Tewari et al. 2020a; Tov et al. 2021; Zhu et al. 2020] aim to solely invert the image, without any modifications while conditional encoders [Alaluf et al. 2021a] are designed for obtaining a latent code conditioned on attributes such as pose, age, or facial expressions. The so-called hybrid methods [Bau et al. 2019b; Zhu et al. 2016] combine optimization-based methods with learning-based methods. The images are first inverted to a latent code by a learned encoder. This latent code then becomes the initialization for the latent optimization, and is optimized to reconstruct the target image.

More recent approaches build different architectures, fine-tune StyleGAN weights, or modulate feature maps for inversion. Style Transformer [Hu et al. 2022] use a combination of convolutional neural networks and transformers to invert images into the latent space. Pivotal Tuning Inversion (PTI) [Roich et al. 2021] fine-tunes the generator around a pivotal latent code to find a balance for the distortion-editability tradeoff. Some methods [Alaluf et al. 2021c; Dinh et al. 2022] train hypernetworks to modulate the weights of a pre-trained StyleGAN network for accurate as well as editable inversions. Spatially-Adaptive Multilayer (SAM) GAN Inversion [Parmar et al. 2022] predicts invertibility maps and High-Fidelity GAN Inversion (HFGI) [Wang et al. 2022] predicts latent maps to modulate StyleGAN features.

While both optimization-based and hybrid approaches may reconstruct images faithfully, they require solving an optimization problem for each image, resulting in longer processing times. On the other hand, our approach adapts learned adapters appended to encoders, which provides a much faster alternative to current methods. Furthermore, we condition the inversion process directly on the target captions, which ensures that a more effective editing space direction can be found in the latent space.

2.3 Latent Space Manipulation

Recent work has shown that GANs learn a semantically-coherent latent space enabling to map manipulations in the latent space to semantic image editing. A common approach is to first invert the input image back into the latent space of a pretrained generator using GAN inversion and then traverse the latent space to find a meaningful direction. Such a direction can be found by either using explicit supervision of image attribute annotations [Abdal et al. 2021; Shen et al. 2020; Wu et al. 2020], or in an unsupervised manner [Härkönen et al. 2020; Shen and Zhou 2021; Voynov and Babenko 2020]. Recently proposed methods consider various modalities for conditional image manipulation. StyleMapGAN [Kim et al. 2021] proposes an intermediate latent space with spatial dimensions with spatial modulation that enables local editing based on reference

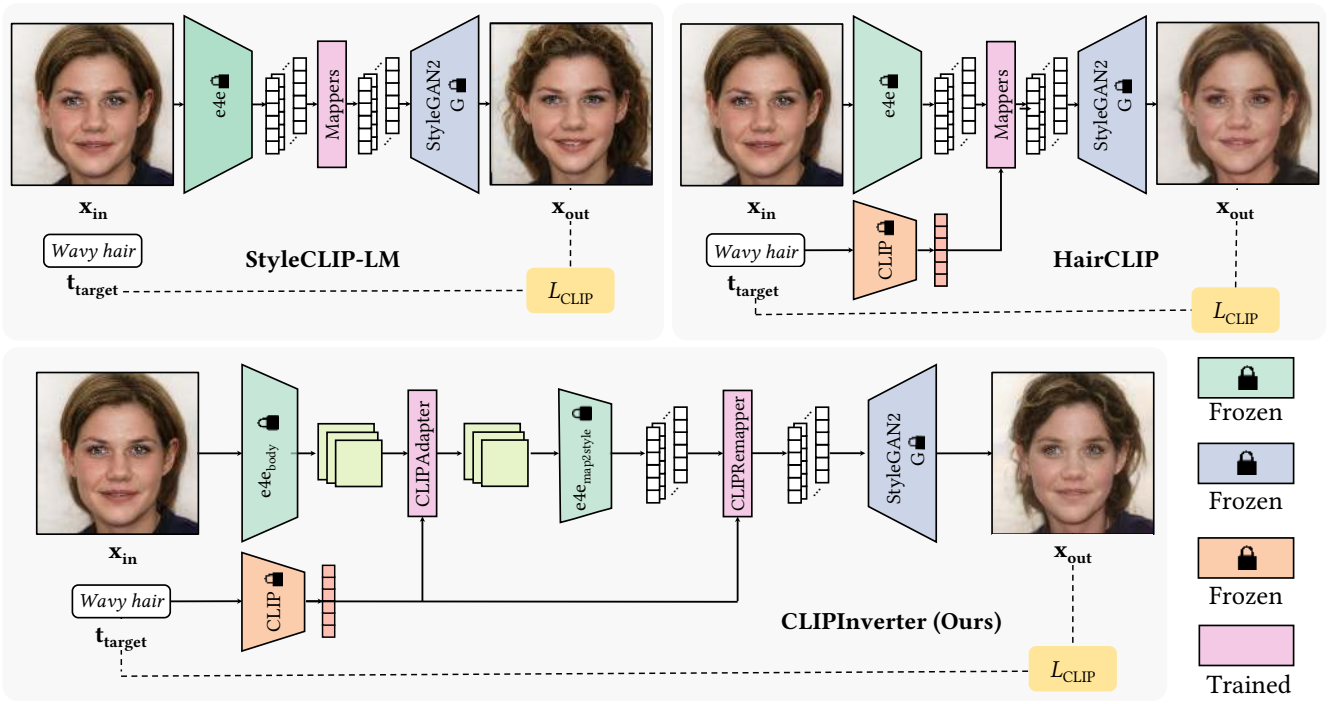


Fig. 2. **An overview of our CLIPInverter approach in comparison to similar text-guided image manipulation methods.** StyleCLIP-LM utilizes target description only in the loss function. HairCLIP additionally uses the description to modulate the latent code obtained by the encoder within the mapper. Alternatively, our CLIPInverter employs specially designed adapter layers, CLIPAdapter, to modulate the encoder in extracting the latent code with respect to the target description. To further obtain more accurate edits, it also makes use of an extra refinement module, CLIPRemapper, to make subsequent corrections on the predicted latent code.

images. Similarly, the study by [Collins et al. 2020] uses a transformation matrix to control the interpolation between an input image and a reference image in the latent space to locally edit the input image. The recent work of [Alaluf et al. 2021a] manipulates an input image based on a target age by training an encoder conditioned on the target age to find residual latent codes to add to the inverted latent code of the original image. In a similar vein, we train adapter layers appended to an encoder conditioned on textual descriptions to output these residual latent codes. We also use the CLIP model to define supervisory signals to explore the similarity of an input image and a textual description.

Moreover, there are several latent spaces to consider in a StyleGAN2 generator. The latent mapper transforms the latent codes in the space $\mathcal{Z}$ drawn from a Normal distribution to an intermediate latent space $\mathcal{W}$. The latent codes in the $\mathcal{W}$ space are used at different stages in the StyleGAN2 generator, after being mapped to the $\mathcal{S}$ space by an affine transformation. $\mathcal{W}+$ space is an extended version of the $\mathcal{W}$ space where a different $\mathbf{w}$ is used for each style input of the generator. While some works find editing directions in the $\mathcal{S}$ space such as StyleCLIP-GD [Patashnik et al. 2021] and StyleMC [Kocasari et al. 2021], many others like StyleCLIP-LO, StyleCLIP-LM [Patashnik et al. 2021], SAM [Alaluf et al. 2021a] utilize the extended intermediate space $\mathcal{W}+$. Our text-guided image encoder operates on $\mathcal{W}+$ to find effective editing directions.

2.4 Text-Guided Image Manipulation

Given an image and a target description in natural language, the aim of text-guided image manipulation is to generate images that reflect the desired semantic changes while also preserving the details or attributes not mentioned in the text. ManiGAN [Li et al. 2020] learns a text-image affine combination module which selects image regions that are relevant to the language description and a detail correction module that modifies these regions. TediGAN [Xia et al. 2021a] enforces the text and image matching by mapping the images and the text to the same latent space and performs further optimization to preserve the identity of the subjects in the original image. In contrast, our method uses a feed-forward text-guided encoder to find residual latent codes and hence is much more efficient.

More recent works use semantics learned by a multi-modal method such as CLIP [Radford et al. 2021]. CLIP learns a joint representation of images and text by mapping image and text pairs into a common latent space. StyleCLIP [Patashnik et al. 2021] uses this space to optimize for the latent code (StyleCLIP-LO) that minimizes the distance of the image & text pair in the CLIP space. They also present a latent mapper (StyleCLIP-LM) that predicts residual latent codes corresponding to specific attributes. Finally, they also experiment with mapping a text prompt to a global direction (StyleCLIP-GD) in the latent space that is independent of the input image. The most recent StyleMC [Kocasari et al. 2021] model presents an efficient method

to learn global directions in the $\mathcal{S}$ space of StyleGAN2 for a given text prompt, by finding directions at lower resolutions and applying manipulations at higher resolutions. It also utilizes CLIP to minimize the distance between the generated image and the text prompt. Most recently and most similar to our approach, HairCLIP [Wei et al. 2022] modulates the inverted latent codes based on hairstyle and hair color inputs as image or text. Their approach is similar to StyleCLIP-LM. However, they also modulate the latent codes with the CLIP embeddings rather than solely optimizing the similarity in the CLIP space.

Our work share some similarities with the aforementioned methods. Like the original TediGAN model, we employ an encoder to predict the latent code conditioned on the provided target description. That said, we estimate a residual latent code reflecting only the desired changes mentioned in the description, which is to be added to the inverted latent code of the input image. StyleCLIP-LM and StyleMC models predict residual latent codes similar to ours, but they require training their mapper functions from scratch for each text prompt via a loss function based on CLIP similarity. Most similar to our approach, HairCLIP applies modulations in the latent space after obtaining inversions with a pretrained network. On the other hand, we let CLIP embeddings modulate the feature maps via an adapter module for predicting the residual latent code. With this modulation, our inversion step is text-guided, whereas HairCLIP applies text-conditioning on the latent space. We also train a correction module which applies latent code blending with learnable blending coefficients for improved accuracy, quality and fidelity in the output images. In Fig. 2, we illustrate the aforementioned fundamental differences between our approach and the most similar StyleCLIP-LM and HairCLIP methods.

Our approach allows us to manipulate fine-scale details by modulating the feature maps, resulting in more accurate manipulations than HairCLIP. Thanks to this process, we also eliminate the need for separate training, unlike StyleCLIP-LM. That is, once our model is trained, it can be directly used to manipulate images by considering a large variety of text prompts containing multiple attributes. We provide extensive comparisons against these recent methods in Section 4 and show the superiority or competitiveness of our proposed approach.

2.5 Diffusion Based Image Generation and Editing

Diffusion models are powerful models that achieve state-of-the-art performance in image generation. These models are trained with variational inference and they synthesize images by gradually removing noise from a signal. Recent diffusion models are capable of generating very high quality images, even surpassing GANs [Dhariwal and Nichol 2021; Ho et al. 2020; Rombach et al. 2022].

With the recent success of the diffusion based models, several conditional diffusion approaches have been proposed for text-guided image generation. VQ-Diffusion [Gu et al. 2021] uses a VQ-VAE and models the latent space by a conditional Denoising Diffusion Probabilistic Model (DDPM) [Ho et al. 2020]. It performs denoising diffusion on discrete image tokens to synthesize text-guided images. GLIDE [Nichol et al. 2021] applies two methods, CLIP [Radford et al. 2021] guidance and classifier-free guidance, to guide the diffusion process with text prompts. Recent large scale text-to-image diffusion models are trained on extremely large datasets containing paired images and texts, like DALL-E2 [et al 2022], Stable Diffusion [Rombach et al. 2021] and Imagen [Saharia et al. 2022], show exceptional performance in generating images from text prompts.

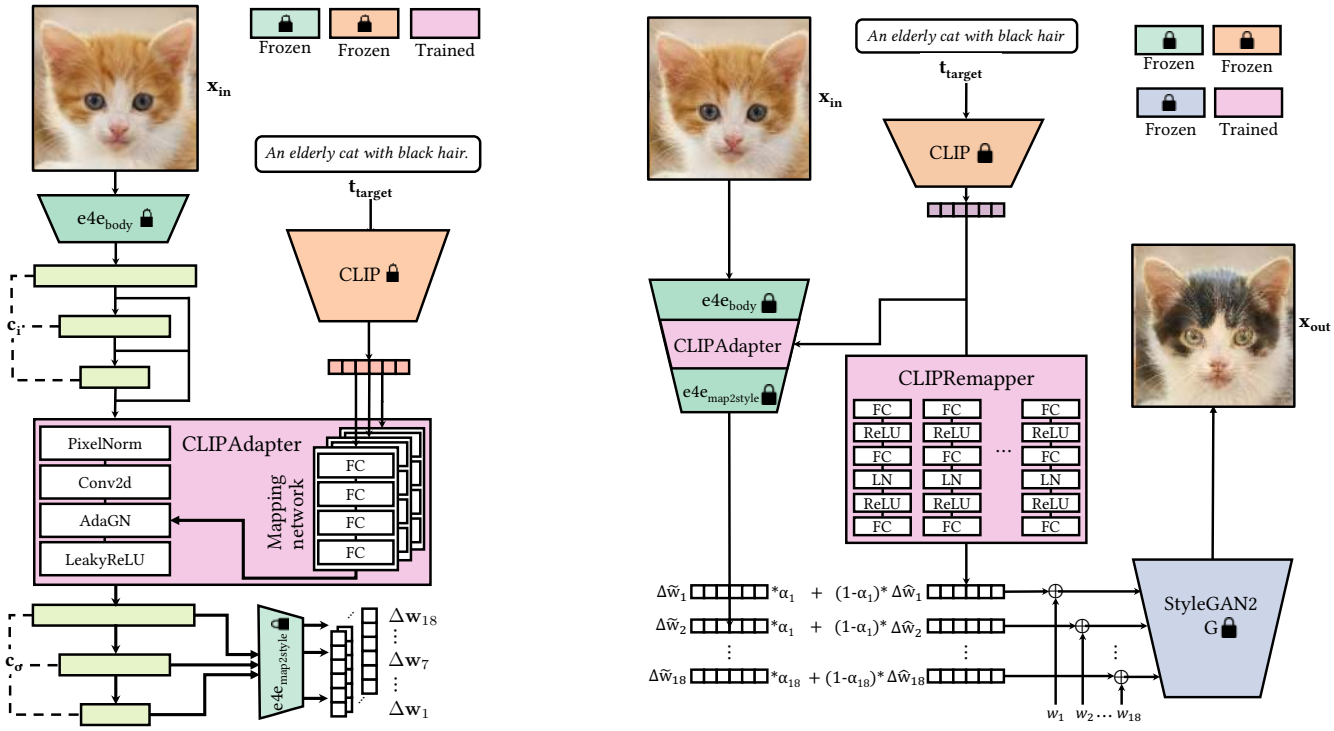
For text-guided image manipulation, DiffusionCLIP [Kim et al. 2022] first converts the images to latent noises by forward diffusion and then guides the reverse diffusion process by CLIP to control the attributes in the synthesized images. UniTune [Valevski et al. 2022] introduces a simple method to fine-tune large scale text-to-image diffusion models on single images. Similarly, Imagic [Kawar et al. 2022] optimizes a text embedding and fine-tunes pretrained generative diffusion models to perform edits on a single image. These diffusion-based editing methods differ from ours as each one requires a large pre-trained text-to-image network.

2.6 Adapter Layers

Adapter layers [Houlsby et al. 2019], originally proposed for NLP tasks, are compact modules that allow parameter sharing in an efficient manner. The key idea is to add adapter modules, consisting of a few layers, between the layers of a pretrained network. The parameters of the adapter module are updated during the fine-tuning phase on a downstream task, while the original parameters of the pretrained network remain the same. This way, most of the parameters of the pretrained network are shared between different downstream tasks, resulting in a model that is able to perform diverse tasks efficiently. Since the parameters of the pretrained network are frozen, the original capabilities of the model are preserved. The module proposed for NLP [Houlsby et al. 2019] is appended after the feed-forward layers and before adding the skip connection back, in a transformer model. This module consists of a down-projection and an up-projection layer. Compared to the original pretrained model, the number of parameters of the adapter module is considerably smaller, allowing the learning of new tasks efficiently.

Adapter layers have also been proposed to use in computer vision tasks. Rebuffi et al. [2017] introduced residual adapter layers for multiple-domain learning in image recognition. Their residual adapter layers are slightly modified versions of the residual blocks in ResNet [He et al. 2016], where batch normalization and 1×1 convolutions with residual connections are added to these residual blocks. [Rebuffi et al. 2018] proposed several improvements over this module. They modified the series implementation of the residual adapter to obtain a parallel adapter, where the input to the convolutional blocks of the residual block is processed in parallel with the adapter convolutions and fed back to the original branch. They also investigated where to place the adapter layers in the ResNet to achieve the best performance. Finally, in VL-Adapter [Yi-Lin Sung 2022], the authors experimented with adapter layers in vision and language joint tasks. They added adapter modules consisting of downsampling and upsampling layers to the transformer architecture for parameter efficient fine-tuning.

Our approach consists of adapter modules that we attach to inversion models. Our encoder adapter module is similar to the mapping networks in StyleCLIP-LM. However, in these adapter modules, we modulate intermediate image feature maps that are extracted from



(a) **Architectural details of CLIPAdapter**. Our text-guided image encoder network inherits the structure of e4e, and makes it conditional on CLIP embedding of the target text. This is achieved by shallow mapping networks at three different scales to better align the multi-modal semantic space of CLIP model with the $\mathcal{W}+$ space of StyleGAN2, whose outputs control the prediction of residual codes through AdaGN layers.

(b) **Architectural details of CLIPRemapper**. Our refinement module consists of MLPs that predict a residual latent code solely based on the CLIP embedding of the target description. This residual is blended with the residual predicted by the CLIPAdapter module, providing a corrected (better aligned) residual latent code which is used to synthesize the final manipulated image via the pre-trained StyleGAN2 generator.

Fig. 3. **CLIPAdapter and CLIPRemapper modules of our CLIPInverter framework.** Our text-guided image editing framework includes two key modules, CLIPAdapter and CLIP Remapper. CLIPAdapter employs CLIP-conditioned adapter layers within the GAN inversion process to find the semantic editing direction in the latent space. CLIPRemapper further refines the predicted edit direction to improve the manipulation accuracy again based on the CLIP embedding of the input text prompt.

the inversion model. After the modulation, the feature maps are fed back to the inversion model to be processed further. With this essential idea, we are able to add a text conditional branch to the existing GAN inversion models while preserving its unconditional inversion capabilities.

3 THE APPROACH

3.1 Overview of CLIPInverter

Our text-guided image editing framework includes two separate modules, namely CLIPAdapter and CLIP Remapper, each playing a different role in obtaining the desired edit. CLIPAdapter involves CLIP-conditioned adapter layers for the GAN inversion process, which are used for finding semantic editing directions in the latent space along which the given input image is manipulated. CLIPRemapper then performs a final refinement over the predicted latent code of the output image considering the CLIP embedding of the input text prompt to further improve the manipulation accuracy as well as the perceptual quality.

Given an input image $\mathbf{x}_{in}$ and a desired target description $\mathbf{t}_{target}$, the goal of our CLIPInverter approach is to manipulate the input image and synthesize an output image $\mathbf{x}_{out}$ such that the end result reflects the attributes described in the text (e.g., hair color, age, gender), while preserving the identity of the subject present in the original image or any other features not relevant to the description. Assuming that we have access to a StyleGAN2 [Karras et al. 2020] generator G that can synthesize images from a particular domain, we cast this text-guided manipulation task as finding a mapping of the input image $\mathbf{x}_{in}$ and the target text prompt $\mathbf{t}_{target}$ to a latent code $\mathbf{w}^* \in \mathcal{W}+$ in the latent space of G so that when decoded it generates the manipulation result as $\mathbf{x}_{out} = G(\mathbf{w}^*)$.

We perform the latent space mapping in two steps, using the unconditional and the conditional branches of the text-guided encoder, which we obtain by attaching CLIPAdapter to a pretrained

image inversion network, namely encoder4editing ($e4e$) [Tov et al. 2021]. We first map the input image $\mathbf{x}_{in}$ to its latent code $\mathbf{w}$ through the pretrained encoder $e4e$. We then compute a residual latent vector $\Delta\mathbf{w}$ through the conditional branch, which processes both the input image and the CLIP model [Radford et al. 2021] embedding of the textual description. The final image $\mathbf{x}_{out}$ is synthesized by passing the aggregated latent code first through the refinement module, $\mathbf{w}^* = CLIPRemapper(\mathbf{w} + \Delta\mathbf{w})$, then through the generator network, which is a pretrained StyleGAN2 [Karras et al. 2020] generator. CLIPInverter applies one final correction to the latent code by predicting latents based on the CLIP embedding of the target caption $\mathbf{t}_{target}$. Then, the predicted latent is blended with the previously inverted latent code depending on a learnt interpolation coefficient α.

In the following, we describe the details of the key modules of CLIPInverter and the loss functions we utilize during training.

3.2 CLIPAdapter: CLIP-Guided Adapters for Latent Space Manipulation

Fig. 3(a) shows the architecture of our proposed text-guided encoder, which follows the architecture of $e4e$ with attached light-weight adapters that enable us to incorporate the textual descriptions. The original $e4e$ architecture maps the input image to feature maps at three levels – coarse, medium and fine. We introduce Adaptive Group Normalization (AdaGN) layers in CLIPAdapter, replacing the Instance Normalization in the AdaIN [Huang and Belongie 2017] layers to modulate these features using features obtained from the CLIP [Radford et al. 2021] embedding of the target textual description.

CLIPAdapter also employs shallow mapping networks, one for each level, to better align the multi-modal semantic space of the CLIP model with the $\mathcal{W}+$ space of StyleGAN2. Specifically, we feed the text embedding obtained from the CLIP model to a multi-layer perceptron (MLP) which predicts the scale and shift parameters of the subsequent AdaGN blocks. Given the image features from the coarse, medium, and fine layers of the encoder, the AdaGN blocks perform feature modulation such that the outputs control the prediction of the residual latent codes.

The design philosophy behind our encoder architecture is to have adapter layers in a pretrained network that can identify visual features relevant and irrelevant to the manipulation task in both image and text-specific manner in computing the residual latent code to identify the manipulation direction in the $\mathcal{W}+$ space.

More formally, in order to manipulate a given image $\mathbf{x}_{in}$ based on a text prompt $\mathbf{t}_{target}$, we start with obtaining the latent code $\mathbf{w}$ of the original image in the $\mathcal{W}+$ latent space of StyleGAN2 [Karras et al. 2020] via $e4e$:

$$\mathbf{w} = e4e(\mathbf{x}_{in}) \in \mathbb{R}^{18\times512}. \tag{1}$$

To perform semantic edits on $\mathbf{x}_{in}$ to reflect the desired target look, we utilize the text-conditioned branch of our encoder network, which takes both the input image and the target textual description as input and outputs the residual latent code. During this process, we first extract intermediate feature maps c_i from the body layers of the encoder network, $e4e_{body}$:

$$\mathbf{c_i} = e4e_{body}(\mathbf{x}_{in}). \tag{2}$$

Next, we utilize the CLIP text embedding of the target text prompt $\mathbf{t}_{target}$ to modulate $\mathbf{c_i}$, obtaining the modulated feature maps $\mathbf{c_o}$ through our encoder-adapter layers CLIPAdapter:

$$\mathbf{c_o} = CLIPAdapter(\mathbf{c_i}, \mathbf{t}_{target}). \tag{3}$$

As the final step to predict the manipulation directions as residual latents $\Delta\mathbf{w}$, we pass the modulated feature maps $\mathbf{c_o}$ through the map2style layers of $e4e$, $e4e_{m2s}$:

$$\Delta\mathbf{w} = e4e_{m2s}(\mathbf{c_o}) \in \mathbb{R}^{18\times512}. \tag{4}$$

Note that the body and map2style layers of $e4e$ combine to complete the pretrained encoder $e4e = [e4e_{body}, e4e_{m2s}]$. The language conditioning happens in the adapter layers CLIPAdapter and these layers are the only layers with trained parameters in the inversion framework, the rest of the parameters are pretrained.

3.3 CLIPRemapper: CLIP-Guided Latent Vector Refinement

In order to further enhance the quality of the manipulated image, we introduce a final refinement step over the predicted latent code. As shown in Fig. 3(b), our CLIPRemapper carries out this refinement process by mapping CLIP text embedding of the given text prompt to the $\mathcal{W}+$ space of StyleGAN and then using the projected text embedding to steer the residual latent code predicted by CLIPInverter towards a direction more compatible with the target text. Specifically, CLIPRemapper involves shallow mapping networks for each level to better align image with the text. The text embedding obtained from CLIP is fed to MLPs at each stage to predict a component for latent code correction corresponding to the caption, as follows:

$$\Delta\hat{\mathbf{w}}_\mathbf{i} = MLP_i(\mathbf{t}_{target}). \tag{5}$$

Taking into account $\Delta\hat{\mathbf{w}}_\mathbf{i}$, we apply a further correction to the residual latent code predicted through CLIPAdapter as:

$$\Delta\mathbf{w}_\mathbf{i}' = \frac{(\alpha_i * \Delta\tilde{\mathbf{w}}_\mathbf{i} + (1-\alpha_i) * \Delta\hat{\mathbf{w}}_\mathbf{i}) * \|\Delta\tilde{\mathbf{w}}_\mathbf{i}\|}{\|\alpha_i * \Delta\tilde{\mathbf{w}}_\mathbf{i} + (1-\alpha_i) * \Delta\hat{\mathbf{w}}_\mathbf{i}\|} \tag{6}$$

where α_i is a weighting factor which is defined as a learnable parameter, and $\Delta\mathbf{w}_\mathbf{i}'$ represents the final corrected residual latent code.

In particular, the corrected residual latent code $\Delta\mathbf{w}_\mathbf{i}'$ is obtained by considering linear combination of two separate codes, the residual latent code from CLIPAdapter $\Delta\tilde{\mathbf{w}}_\mathbf{i}$ and the vector $\Delta\hat{\mathbf{w}}_\mathbf{i}$, followed by a normalization. We do not want the refinement procedure to make substantial changes in the predicted latent code. Hence, along with the loss functions introduced in the next section, the normalization further enforces the final latent code $\Delta\mathbf{w}_\mathbf{i}'$ to be in the vicinity of the residual latent code predicted in the previous step. We only make the necessary changes in the semantic directions suggested by the CLIP embedding of the target text $\mathbf{t}_{target}$ through a simple image composition process in the latent StyleGAN space.

As demonstrated in Fig 4, in a way, this refinement process, composes the manipulated image obtained by CLIPAdapter with a generic image mostly having the characteristics mentioned in the target description, leading to further improvements on both the manipulation accuracy and the perceptual quality.

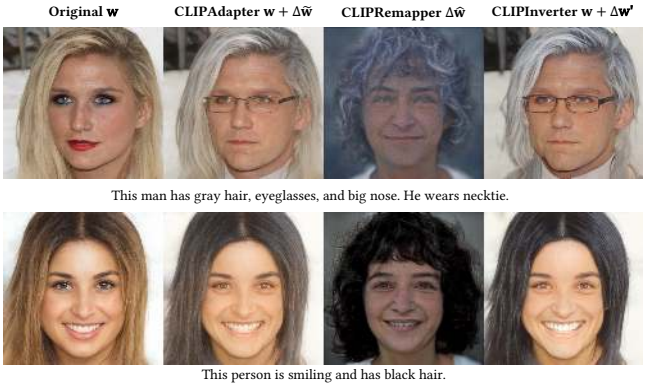


Fig. 4. **Visualization of the latent code correction operation via CLIPRemapper.** For two sample images, we show the initial editing results generated solely by CLIPAdapter, the generic images generated via CLIPRemapper, and the final manipulations by CLIPInverter obtained by the suggested correction scheme. Our refinement module works as intended, providing edits more consistent with the target descriptions.

3.4 Training Losses

We train our proposed encoder-adapters, CLIPAdapter, and the refinement module, CLIPRemapper, on a training set of images paired with their corresponding textual descriptions $\{(\mathbf{x}_{in}, \mathbf{t}_{real})\}$. Specifically, we employ a cyclic adversarial training strategy [Zhu et al. 2017] during training, i.e. we perform two separate manipulations. In the first one, we feed in the original input image $\mathbf{x}_{in}$ together with a target textual description $\mathbf{t}_{target}$ that does not match the input image to our model. The output of this first pass is the manipulated image $\mathbf{x}_{out} = \text{CLIPInverter}(\mathbf{x}_{in}, \mathbf{t}_{target})$. In the cyclic pass, we feed this manipulated image $\mathbf{x}_{out}$ and the original text description $\mathbf{t}_{real}$ (the description of the original input image $\mathbf{x}_{in}$) to obtain $\widehat{\mathbf{x}}_{in} = \text{CLIPInverter}(\mathbf{x}_{out}, \mathbf{t}_{real})$. We expect $\widehat{\mathbf{x}}_{in}$ to resemble the original image $\mathbf{x}_{in}$ by enforcing cycle consistency. We obtain the target text description by rolling the minibatch, meaning that each image will be paired with the textual description that describes the next image in the minibatch.

We train our model by using a loss function that consists of 5 different objectives. Each of these objectives are used both in the first manipulation pass and the following cycle pass. Here, we describe our losses for only the first manipulation pass for simplicity.

We use $\mathcal{L}_2$ and $\mathcal{L}_{\text{LPIPS}}$ [Zhang et al. 2018] losses to respectively enforce pixel-wise and perceptual similarities between the input and the manipulated image, such that:

$$\mathcal{L}_2 = \|\mathbf{x}_{in} - \mathbf{x}_{out}\|_2\,, \tag{7}$$

$$\mathcal{L}_{\text{LPIPS}} = \|F(\mathbf{x}_{in}) - F(\mathbf{x}_{out})\|_2\,, \tag{8}$$

where $F(\cdot)$ denotes deep features extracted from a pretrained AlexNet [Krizhevsky et al. 2012] model.

Ideally, we want any manipulation to preserve the identity of the subject in the original image. To preserve the identity, we employ an identity loss which maximizes the cosine similarity between the input image and the output image feature embeddings:

$$\mathcal{L}_{\text{ID}} = 1 - \langle R(\mathbf{x}_{in}), R(\mathbf{x}_{out})\rangle\,, \tag{9}$$

where $\langle\cdot,\cdot\rangle$ represents the cosine similarity between the feature vectors, R denotes a pretrained deep network. Specifically, we use the pretrained ArcFace [Deng et al. 2019] network for human faces, and a ResNet50 [He et al. 2015] network trained with MOCOv2 [Chen et al. 2020] for birds and cats.

We also employ the following regularization loss, which enforces the predicted latent codes to be close to the average latent code of the generator, and shown to improve overall image quality in previous work work [Richardson et al. 2021], such that:

$$\mathcal{L}_{\text{reg}} = \|\mathbf{w}^* - \overline{\mathbf{w}}\|_2\,, \tag{10}$$

where $\mathbf{w}^*$ is the aggregated latent code and $\overline{\mathbf{w}}$ is the average latent code.

Lastly, to enforce the similarity between the output image and the target description, we employ a directional CLIP loss [Gal et al. 2021]. Rather than directly minimizing the distance between the generated image $\mathbf{x}_{out}$ and the text prompt $\mathbf{t}_{target}$ in the CLIP space, directional CLIP loss aligns the direction from the input image $\mathbf{x}_{in}$ to the manipulated image $\mathbf{x}_{out}$ with the direction from the original text description $\mathbf{t}_{real}$ to the target text description $\mathbf{t}_{target}$:

$$\begin{aligned}\Delta T &= E_{\text{CLIP,T}}\left(\mathbf{t}_{target}\right) - E_{\text{CLIP,T}}\left(\mathbf{t}_{real}\right),\\ \Delta I &= E_{\text{CLIP,I}}\left(\mathbf{x}_{out}\right) - E_{\text{CLIP,I}}\left(\mathbf{x}_{in}\right),\\ \mathcal{L}_{\text{direction}} &= 1 - \frac{\Delta I \cdot \Delta T}{|\Delta I||\Delta T|},\end{aligned} \tag{11}$$

where $E_{\text{CLIP,T}}$ and $E_{\text{CLIP,I}}$ are the text and image encoders of CLIP, respectively.

Our final loss function for the first manipulation pass is a weighted sum of the objectives:

$$\mathcal{L}_{\text{manipulation}} = \lambda_1\mathcal{L}_2 + \lambda_2\mathcal{L}_{\text{LPIPS}} + \lambda_3\mathcal{L}_{\text{ID}} + \lambda_4\mathcal{L}_{\text{reg}} + \lambda_5\mathcal{L}_{\text{direction}}\,, \tag{12}$$

where each λ_i determines the weight of the corresponding objective. The total loss including the first manipulation and the follow-up cycle passes is the following:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{manipulation}} + \lambda_6\mathcal{L}_{\text{cyclic}}\,, \tag{13}$$

where $\mathcal{L}_{\text{cyclic}}$ is the cyclic consistency loss, which contains the same loss terms as $\mathcal{L}_{\text{manipulation}}$ and λ_{cycle} is the weight for this cyclic loss.

During training, we follow a multi-stage regime. We first train the CLIPAdapter (without using CLIPRemapper). Once these are fully trained, we freeze the weights of CLIPAdapter weights and train the CLIPRemapper while optimizing for the CLIP loss along with the L2, LPIPS and ID losses. For the LPIPS and L2 losses, we also include the loss between images generator with and without CLIPRemapper which ensures that the CLIPRemapper does not changes the images by a large amount. In addition, we also include a L2 regularization loss on the interpolation coefficients (lambdas) such that the amount of interpolation between two latent codes does not change the original code by a large amount. This is also observed to remove artifacts in the generated images.

4 EXPERIMENTAL EVALUATION

4.1 Datasets

We conduct extensive evaluation on a variety of domains to illustrate the generalizability of our approach. We use the Multi-Modal CelebA-HQ [Lee et al. 2020; Xia et al. 2021a] dataset to train our model on the domain of human faces. This dataset consists of 30,000 images along with 10 textual descriptions for each image. We follow the default train/test split, using 6000 images for testing and the remaining for training. For the birds domain, we use the CUB Birds dataset [Wah et al. 2011], which contains 11,788 images in total, including 2933 images for testing, along with 10 captions for each image. Finally, for the domain of cat faces, we use the AFHQ-Cats dataset [Choi et al. 2020] which contains a total of 5653 images, including 500 for testing. The captions for this dataset are generated using the approach mentioned in [Nie et al. 2021] leveraging the CLIP [Radford et al. 2021] model.

4.2 Training Details

We use two pre-trained models trained on our datasets: StyleGAN2 generator and *e4e* encoder. Keeping the weights of these models frozen, we train CLIPInverter using the cyclic adversarial training scheme described in the previous section. The mismatching captions are sampled in such a way that matching caption for an image is sampled 25% of the time during training. In our experiments, for the CLIPAdapter, we empirically set $\lambda_1 = 1.0$, $\lambda_2 = 0.6$, $\lambda_3 = 0.1$, $\lambda_4 = 0.005$, $\lambda_5 = 1.0$, $\lambda_6 = 1.0$ and the learning rate to 0.0005. For CLIPRemapper, we increase the weight of the identity loss to $\lambda_3 = 0.5$, and totally exclude the regularization loss during training. We initialize the linear coefficient α_i's with 0.05 and train them together with the parameters of CLIPRemapper. We train CLIPAdapter for 200k iterations on a single Tesla v100 GPU, which takes about 6 days and CLIPRemapper for 20k iterations which takes about a day.

4.3 Evaluation Metrics

Quantitative analysis of the language-guided image manipulation task is a challenging matter. The quality and the photorealism of the generated images can be evaluated with Fréchet Inception Distance (FID) [Heusel et al. 2017]. However, there is no established way to evaluate the manipulation accuracy of a model. It is crucial that an effective model should only alter the attributes specified in the target text prompt, while preserving the original attributes for the rest of the input image.

To evaluate the model accuracy in terms of these aspects, we propose two metrics: Attribute Manipulation Accuracy (AMA) and CLIP Manipulative Precision (CMP). Attribute Manipulation Accuracy measures how accurately a model can apply single attribute manipulations. For face images, we train an attribute classifier using the images and their attribute annotations from the CelebA [Liu et al. 2015] dataset, following [Nie et al. 2021]. In terms of the validation accuracy of the classifier on different attributes, we select 15 of the best performing attributes, such as *blond hair*, *chubby*, *mustache* (see the appendix for the full list of attributes), out of 40 that are included in CelebA. Here, we have two versions of the AMA score. AMA-Single corresponds to the accuracy for single attribute manipulations, where we generate 50 single attribute manipulations using that model with pre-defined text prompts based on the corresponding attribute, such as *"This person has blond hair"*, for each of the 15 selected attributes (750 images in total). We evaluate the accuracy of these manipulations using the attribute classifier and take the mean of the accuracy across all the attributes to obtain the final AMA score for that model. In AMA-Multiple, we consider multiple attribute manipulations, in which the target description involves combinations of two or three attributes, we treat the resulting manipulation successful only if the corresponding attribute classifiers can correctly classify the resulting changes.

For cat and bird images, we use CLIP as a zero-shot classifier to calculate the AMA. We employ 30 attributes present in the AFHQ-Cats [Choi et al. 2020] and sample 40 attributes out of the 273 attributes present in the CUB [Wah et al. 2011] dataset. For each selected attribute, we generate template based captions covering all the classes in the category that the attribute belongs to. Then, we prompt CLIP with the output image and the generated captions to obtain similarity scores for each caption. The manipulation then is successful if the caption with the correct label has the highest probability after the softmax operation on the similarity scores.

CLIP Manipulative Precision is a modified version of the Manipulative Precision metric proposed by ManiGAN [Li et al. 2020] that uses the pre-trained CLIP [Radford et al. 2021] image and text encoders. CMP measures how aligned the synthesized image is with the target text prompt $\mathbf{t_{target}}$ and how well the original contents of the input image are preserved. It is defined as

$$\text{CMP} = (1 - \text{diff}) * \text{sim}, \tag{14}$$

where diff is the $\mathcal{L}_1$ pixel difference between the input image $\mathbf{x_{in}}$ and the output image $\mathbf{x_{out}}$, and sim is the CLIP similarity between the output image $\mathbf{x_{out}}$ and the target textual description $\mathbf{t_{target}}$. We calculate the CMP for each of the images generated for the AMA score and take their average to obtain the final CMP score for the corresponding model.

4.4 Qualitative Results

In Fig. 5, we show that our method can manipulate images from very different domains such as human faces, cats, and birds. Given an input image, we manipulate it by just providing a natural textual description highlighting the desired edits. As can be seen in the figure, the target descriptions can specify more than one attribute. For instance, one can simultaneously apply lipstick while changing the hair style of a woman, or can alter the attitude and appearance of a cat at the same time.

Our method can give plausible results independent of the complexity of the provided target description. For instance, in Fig. 6, we present the outcomes of our approach obtained by taking into account compositions of different visual attributes. They demonstrate that our method can deal with the provided compositions, and make the necessary changes in the original input images mentioned in the descriptions to its full extent.

In Fig. 7, we demonstrate that predicting residual latent code for a given target description has the advantage that one can continuously interpolate between the original image and the final result, which allows users to have control over the degree of changes made

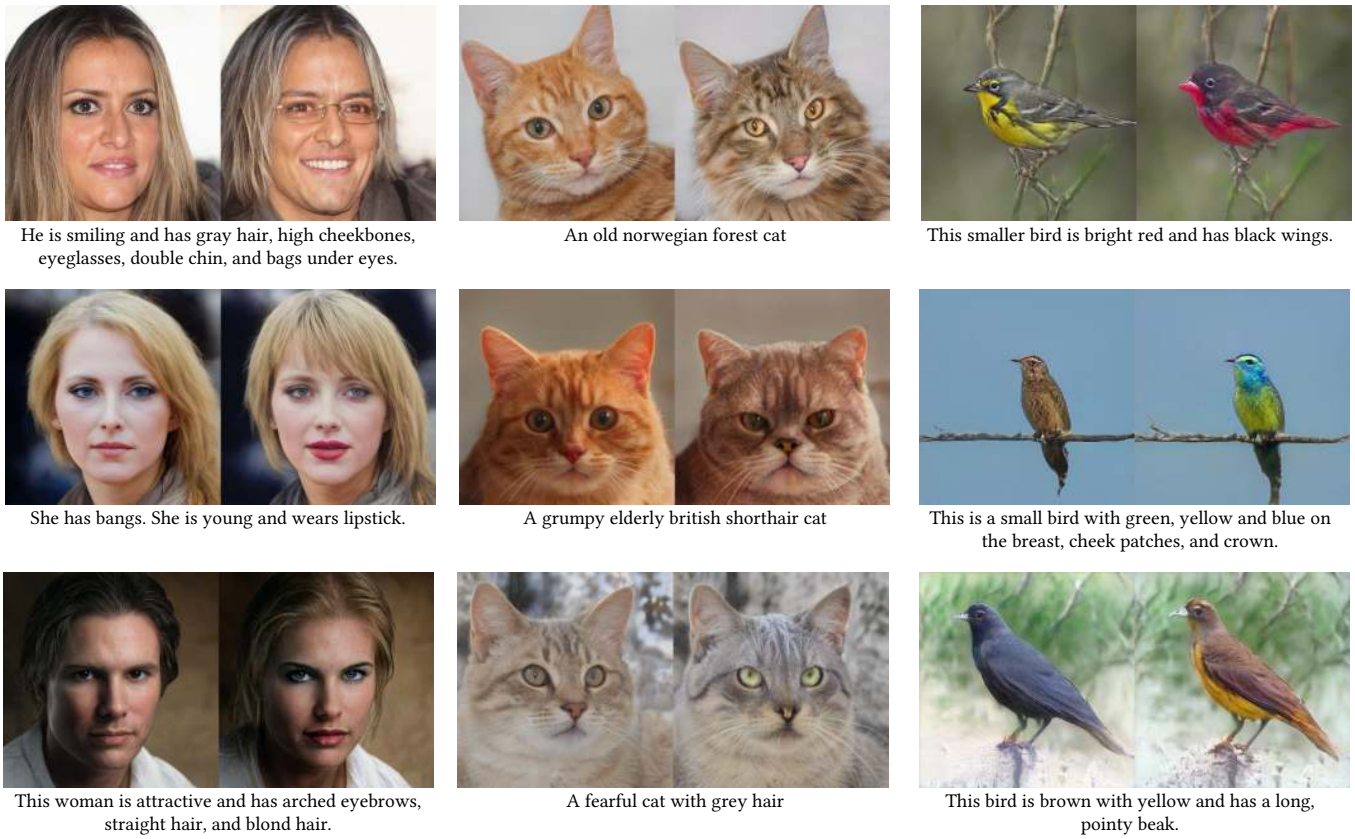


Fig. 5. **Qualitative manipulation results.** We show sample text-guided manipulation results on human faces (*left*), cat images (*middle*), and bird images (*right*). Our approach successfully makes local semantic edits based on the target descriptions while keeping the generated outputs faithful to the input images.

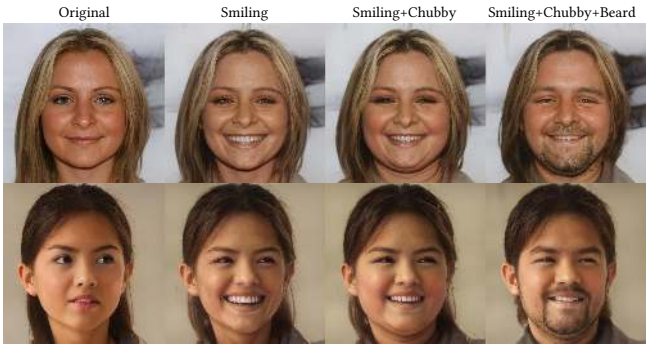


Fig. 6. **More qualitative results.** We provide example manipulation results where we apply various compositions of several facial attributes as target descriptions.

during the manipulation process. For example, the appearance of the subjects smoothly changes to reflect the increase in the intensity of the lipstick, and the color of the cats and the bird slightly changes.

To some extent, our approach can also perform edits in a zero-shot setting by using descriptions never seen during training. The key to this ability lies in the use of the CLIP-based text guided adapters which enable to align the visual and the textual domains and map out of domain textual descriptions to a semantic editing direction in the latent space. Hence, even the terms in the target descriptions have not been observed for the first time, our method can make the necessary changes in the input images if semantically similar terms have been seen during training. For instance, in Fig. 9, we include a number of cases where the color or the structure of the hair is manipulated using novel descriptions that do not exist in the training set such as *curly hair*, *silver hair*, and *facial hair*.

In our proposed CLIPAdapter, we employ CLIP embeddings of the text prompt to modulate the convolutional feature maps to predict the residual latent code, representing the changes on the input image required to meet the desired target description. In fact, CLIP model learns the alignment between images and text via a contrastive learning objective and discovers a common semantic space. Hence, our framework also allows for using exemplar images as the conditioning element without any changes or training. In Fig. 8, we provide some qualitative results for such image-based

Fig. 7. **Continuous manipulation results.** We show that starting from the latent code of the original image and walking along the predicted residual latent codes, we can naturally obtain smooth image manipulations, providing control over the end result. For reference, we provide the original (*left*) and the target descriptions (*right*) below each row.

manipulations performed by our proposed approach. We observe that although no further training is done by considering reference images instead of target description, our model achieves a good performance on transferring the appearance of the provided reference images to the input images.

We refer readers to the supplementary material for more manipulation results.

4.5 Qualitative Comparisons to Other Text-guided Manipulation Methods

We compare our approach with various existing methods, including TediGAN [Xia et al. 2021a], StyleCLIP [Patashnik et al. 2021], StyleMC [Kocasari et al. 2021] and HairCLIP [Wei et al. 2022]. For StyleCLIP, we use the latent optimization based model StyleCLIP-LO, and for TediGAN, we use the CLIP-based optimization approach (TediGAN-B). In all of our experiments, we use the public implementations provided by the authors. For HairCLIP, we slightly modify its neural architecture and train it accordingly. In the original paper, they do consider different conditioning vectors for the mapper modules encoding hairstyle and hair color as they refer to details from different scales. Since, we focus on a generic text-guided manipulation process where it is hard to separate the textual terms into fine, mid and high-level attributes, we let the embedding of the whole target description suggested by CLIP text encoder to condition the

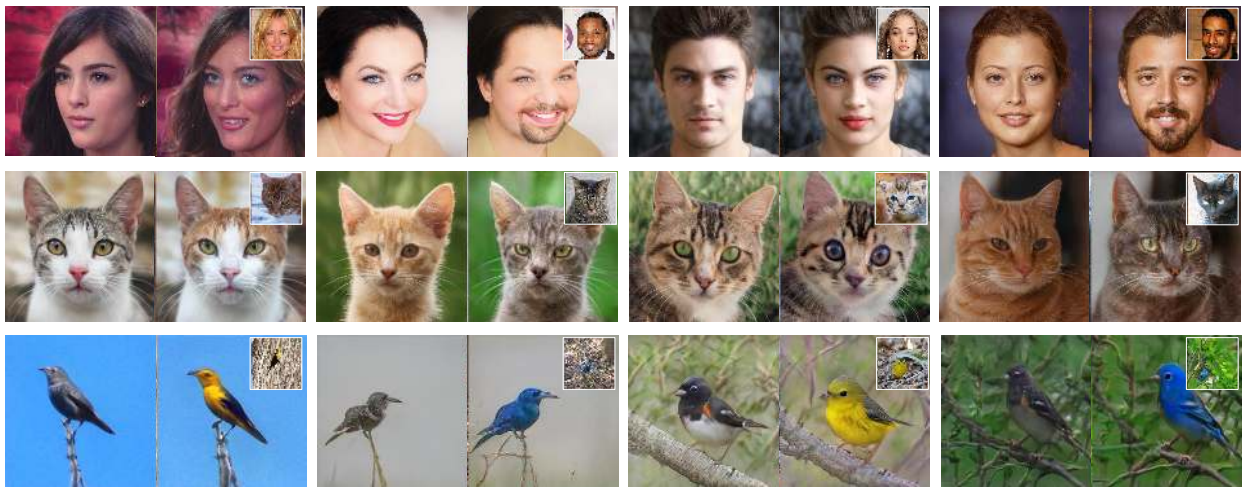


Fig. 8. **Image-based manipulation results.** Our framework allows for using a reference image as the conditioning input for editing. In the figure, these reference images are given at the top-right. Results on different domains illustrate that our model can transfer the look of the conditioning images to the provided input images.

Fig. 9. **Additional manipulation results with out-of-distribution training data.** We demonstrate that our Text2Style method can perform manipulations with target descriptions involving words never seen during training but semantically similar to the observed ones.

mappers equally. All of these approaches use StyleGAN2 as a frozen generator and utilize the CLIP embedding to measure the image and text similarity.

In Fig. 10, we provide some qualitative comparisons between our method and the baselines on a number of human face images. As can be seen from the figure, our approach gives more accurate edits as compared to the existing methods, especially for captions that describe multiple attribute manipulations. For instance, for the first image, our model is able to make meaningful changes to the original input image to reflect the look depicted in the target description, and apply the gender change as well as changes in the eyebrows, hair, eyes, lips and the outfit. For the second input image, our model is able to generate the smile and the lipstick while most of the other methods fail to apply both changes at the same time. In the last two examples, our manipulation results again reflect the given target descriptions – much better than those of the competing approaches. Our method manipulates the gender, hair color, eyebrows, age of the man and applies makeup. Similarly, it generates a smile for the woman and makes her wear a jacket, which is inline with the necktie mentioned in the description. Similarly, in Fig. 11, we compare our results with those of the TediGAN-B, StyleCLIP-LO and HairCLIP methods on bird and cat images. Like the human faces, our model is able to generate visually more pleasing and relevant results than the competing approaches. For instance, our model is able to capture the yellow-greenish color mentioned in the description for the bird in the third row and the fearful look for the cat in the first row while other methods result in poor manipulations. For birds and cats, we could not provide any comparison against StyleCLIP-GD and StyleMC as their codebase use a different implementation of the StyleGAN and they do not provide pre-trained models for these datasets. In the supplementary material, we provide additional visual comparisons.

4.6 Quantitative Comparisons to Other Text-guided Manipulation Methods

We quantitatively compare our approach to the same approaches that are compared in the qualitative comparisons, namely TediGAN [Xia et al. 2021a], StyleCLIP-LO and StyleCLIP-GD [Patashnik et al. 2021], StyleMC [Kocasari et al. 2021] and HairCLIP [Wei et al. 2022]. We use the three metrics mentioned in Section 4.3 (Fréchet Inception Distance (FID), Attribute Manipulation Accuracy (AMA) and CLIP Manipulative Precision (CMP)) for these quantitative comparisons. The official PyTorch implementation [Seitzer 2020] is utilized to calculate the FID scores. The AMA and the CMP scores are calculated using the procedure described in Section 4.3.

Table 1 shows the quantitative comparisons for our model against various state-of-the-art approaches. TediGAN-B achieves fairly good FID and CMP scores. However, from the qualitative results, we observed that TediGAN-B exploits adversarial ways to optimize

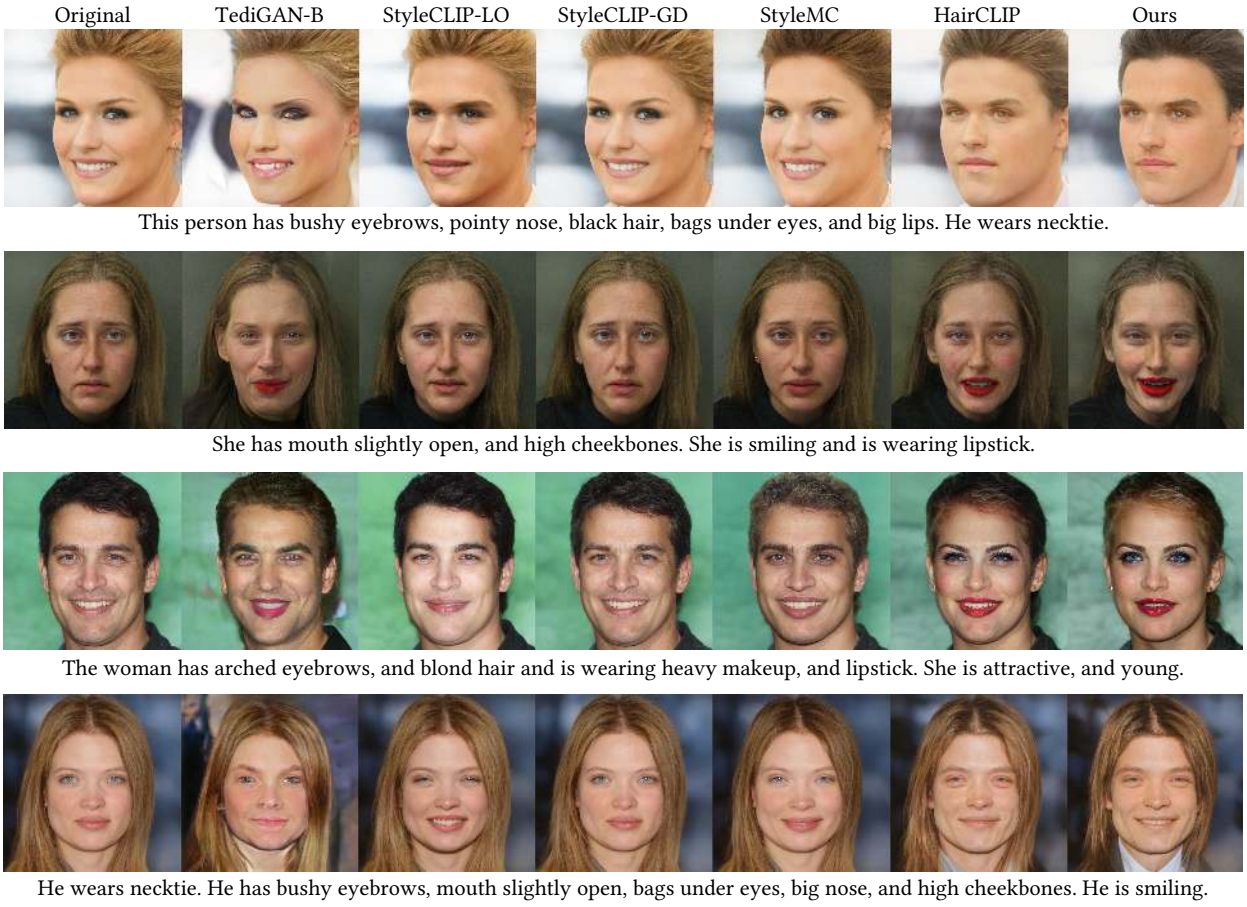


Fig. 10. **Comparison against the state-of-the-art text-guided manipulation methods.** Our method applies the target edits mentioned in the given descriptions much more accurately than the competing approaches, especially when there are multiple attributes present in the descriptions.

the CLIP similarity without changing the input pixels much while failing to apply the manipulations and producing distorted images.

While performing well in terms of either one or two metrics, the competing approaches usually fail to be competitive across all three metrics. StyleCLIP-LO is able to achieve a fairly comparable CMP, since it optimizes the CLIP similarity for each instance, and a good FID score but fails to apply the given attribute manipulations accurately. StyleMC also achieves a good FID score since it finds directions in the $\mathcal{S}$ space. However, it also fails to output accurate manipulations. Even though StyleCLIP-GD performs better than these two models, its performance still falls behind the performance of our approach. Finally, HairCLIP achieves the best scores out of the competing approaches. The results demonstrate the superiority of our model against HairCLIP, as our method achieve much higher manipulation accuracies while remaining competitive in terms of the FID score. Our approach finds a good balance for the distortion and editability problem by applying manipulations successfully while being comparable in terms of photorealism. Hence, they are able to achieve good scores across all three metrics.

Table 1. **Quantitative comparisons on the CelebA dataset.** Our approach demonstrates superior manipulation accuracy compared to other methods, especially for the manipulations involving multiple attributes while also preserving a comparable perceptual quality. The best the second best performing models are shown in bold typeface and underlined, respectively.

	FID ↓	CMP ↑	AMA (Single) ↑	AMA (Multiple) ↑
TediGAN-B	**55.424**	**0.285**	11.286	1.142
StyleCLIP-LO	<u>80.833</u>	0.210	15.857	3.429
StyleCLIP-GD	82.393	0.191	33.143	11.429
StyleMC	84.088	0.187	12.143	2.857
HairCLIP	93.523	0.218	<u>41.571</u>	<u>15.149</u>
Ours	97.210	<u>0.221</u>	**61.429**	**41.714**

Table 2 presents the quantitative comparisons on the AFHQ-Cats and the CUB datasets. Since CLIP is used as a similarity metric in CMP and as a zero-shot classifier in AMA estimations, TediGAN-B again achieves really good scores in these two metrics. However, as seen from the FID scores and the results shown in Fig. 11, it

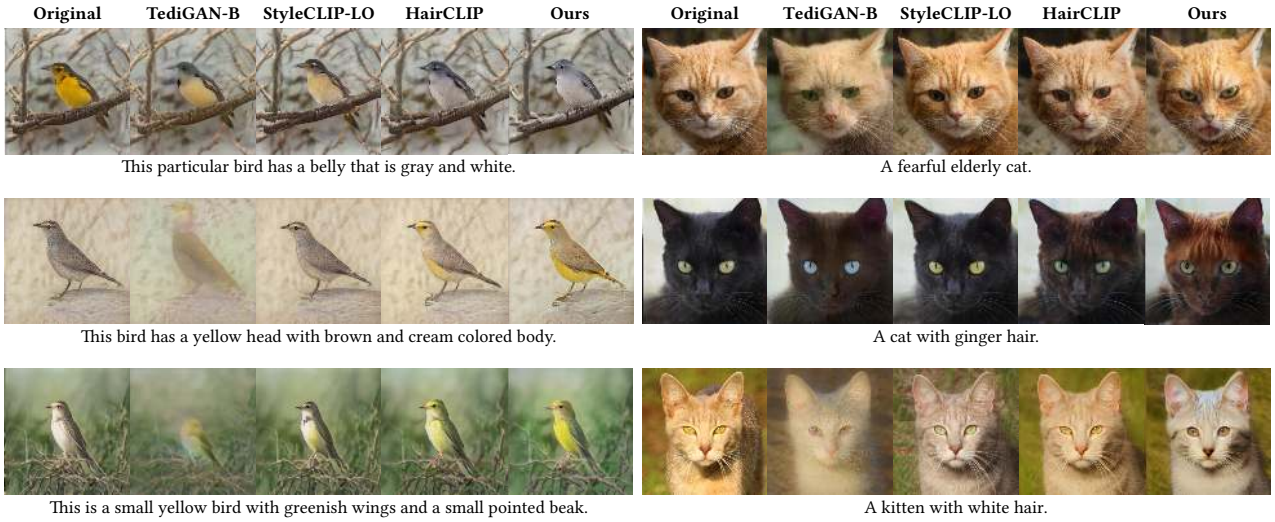


Fig. 11. **Comparisons against TediGAN on bird and cat images.** As compared to TediGAN, our model generates reasonable manipulation results which are more consistent with the given target descriptions.

Table 2. **Quantitative comparisons on the AFHQ-Cats and CUB datasets.** Our approach demonstrates superior manipulation accuracy compared to other methods, while also preserving a comparable perceptual quality. The best the second best performing models are shown in bold typeface and underlined, respectively.

	AFHQ-Cats			CUB		
	FID ↓	CMP ↑	AMA ↑	FID ↓	CMP ↑	AMA ↑
TediGAN-B	39.414	**0.255**	**82.467**	42.007	**0.233**	59.500
StyleCLIP-LO	**18.771**	0.226	48.133	**19.209**	0.211	27.000
HairCLIP	21.087	0.227	44.667	26.447	0.218	57.050
Ours	24.172	0.245	76.467	25.837	0.221	**66.000**

gives highly blurred and non-realistic outputs that are not actually in line with the target descriptions. Another optimization based method, StyleCLIP-LO, achieves worse AMA and CMP scores than TediGAN-B, but better FID. Their loss functions allow the model to output realistic outputs, but they fail to apply the manipulations successfully, which can be seen in Fig. 11. HairCLIP generates images that are better in line with the descriptions than the aforementioned methods. However, our approach outperforms HairCLIP by a large margin in terms of CMP and AMA while having a fairly close or even better FID values. We underlined the second best performing models for each metric, to demonstrate the superiority of our approach against the others, since the best performing models usually exploit adversarial ways to optimize the CLIP similarity which yield high CMP and AMA values or fail to apply the manipulations which yield better FID values.

For quantitative analysis, we conduct a user study via Qualtrics to evaluate the performance of our approach and all the other competing methods. Specifically, in this user study, we focus on two important aspects: (1) the accuracy of the edits with respect to the given target descriptions, and (2) the photorealism of the manipulated images. In our human evaluation, we randomly generate 48 questions, and divide them into 3 groups, with 16 questions each. We make sure that at least 14 different subjects answer each of these group of questions. To measure the accuracy, we show the users an input image, a target description, and the manipulation results of all of the competing methods, and ask them to rank the results against each other with respect to how consistent the edits are to the provided description. The participants perform this by dragging the images into their preferred order, where the left-most position refers to the worst result having rank order 1 and the right-most one represents the best outcome at rank order 6. In order to avoid any bias in the evaluation, the outputs of the methods are displayed in random order at each time. For the questions regarding photorealism, we design a similar ranking task, but this time, we show all the results in random order, and ask the participants to order these results with respect to how realistic they look. Please refer to the supplementary for a screenshot of our user study given to the participants.

Table 3. **User study results.** The table represent the average rankings of the methods with respect to accuracy and realism, where the higher the value is the better the method is. The participants favor the results of our proposed model over the current state-of-the-art when the accuracy of the manipulations is considered.

Task	TediGAN-B	StyleCLIP-LO	StyleMC	StyleCLIP-GD	HairCLIP	Ours
Acc.	1.848	3.401	3.526	3.611	4.015	4.598
Real.	1.218	4.604	4.282	3.609	3.544	3.743

Table 3 summarizes the results of our study where the average ranking scores are reported. We find that in terms of the accuracy, the human subjects prefer our proposed method against all the competing approaches. That is, our method makes only the necessary

edits in the input images with respect to the given target descriptions in a precise manner. HairCLIP and StyleCLIP-GD give the next most accurate results following our model. In terms of photorealism, our results are also superior than these two approaches, indicating that our results are both accurate and photo-realistic. That said, the human subjects find the photorealism of the results of the concurrent StyleMC and StyleCLIP-LO models significantly better. However, the accuracy questions indicate that both StyleMC and StyleCLIP-LO have difficulty in manipulating the given input images in regard to the target descriptions, in contrast to our proposed model. StyleMC and StyleCLIP-LO, in general, make minimal, mostly insufficient changes in the input images (as also can be seen from Fig. 10), and thus do not degrade the photorealism much.

4.7 Ablation Study

During training our model, we leverage different loss terms. In order to analyze the contributions of these loss terms, we have performed an ablation study where we either remove or modify some of these loss terms during training. We provide visual comparisons between these models separately trained on different loss terms in Fig 12.

Firstly, we employ the directional CLIP loss following [Gal et al. 2021], to better enforce the image and description similarity. Compared to the global CLIP loss, which directly minimizes the distance between the manipulated image $\mathbf{x}_{out}$ and the text prompt $\mathbf{t}_{target}$ in the CLIP space, the directional CLIP loss aligns the directions between the real and target descriptions and input and output images. As can be seen in the second column of Fig 12, the global CLIP loss suffers from artificial-looking manipulations and results in poorly constructed facial attributes as compared to the directional CLIP loss.

Secondly, in order to preserve the features and the details of the input image in the areas that we do not wish to modify, we employ the perceptual $\mathcal{L}_2$ and the $\mathcal{L}_{\text{LPIPS}}$ losses between the input and the output images. In theory, these perceptual loss terms contradict the directional CLIP loss since the CLIP loss is trying to enforce the image & text similarity by manipulating the pixel values. In order to analyze the contribution of these perceptual terms, we have reduced the weights of these loss terms in the overall objective. The third column in Fig. 12 shows a manipulation example from this experiment. As can be seen, the smile in the first row is also modified, and the model manipulates the hair style to curly hair in the second row even though this manipulations were not mentioned in the target description. This experiment demonstrates the necessity of these perceptual loss terms in order to prevent unwanted manipulations.

Thirdly, we employ a cyclic-adversarial training strategy, where we first manipulate the image with a mismatching caption, and then recover it by manipulating the output of the first pass with the matching target description. The fourth column in Fig. 12 shows an example manipulation from the experiment where we remove this cyclic training regime. Even though the output is visually similar to the output from our full model, we observe that the cyclic consistency loss helps with the preservation of the identity as well as the manipulation accuracy.

Finally, we utilize a CLIP-guided correction module CLIPRemapper to apply the manipulations more accurately and increase the image fidelity. We see from the last two columns of the figure that without CLIPRemapper, the model is not able to apply all of the specified manipulations accurately, like the hair color in the first row or the earrings in the second row.

Table 4. **Quantitative analysis of the ablation study.** We have performed a quantitative analysis of the ablation study where we calculated the metrics for each of the described experiments for our model. The results demonstrate that our model finds a good balance for applying manipulations without decreasing the perceptual quality of the generations. The best the second best performing models are shown in bold typeface and underlined, respectively.

	FID ↓	CMP ↑	AMA ↑
Ours w/ Global CLIP Loss	**83.404**	**0.221**	25.429
Ours w/o Perceptual Losses	105.432	0.194	**65.571**
Ours w/o Cycle Pass	<u>85.851</u>	0.215	40.857
Ours w/o CLIPAdapter	89.244	0.202	41.571
Ours w/o CLIPRemapper	88.395	<u>0.216</u>	53.28
Ours	97.210	**0.221**	<u>61.429</u>

Table 4 shows the quantitative analysis of the experiments described above. The metrics verify that the global CLIP loss performs much worse in terms of attribute manipulations. This model is able to achieve a high CMP since it directly optimizes image and text similarity in the CLIP space, rather than aligning semantic directions. When we reduce the weight of the perceptual losses, the model is able to apply the manipulations with very high accuracy. However, this comes with the price of perceptual quality as FID suggests, and unwanted manipulations as CMP suggests. Adding the cycle pass gives us a better supervision signal to train the model, as the improvements in the accuracy and the CMP suggest. Without CLIPAdapter, our model is not able to achieve great accuracy scores, suggesting that the adapter layers yield the residual latent codes corresponding to semantic directions successfully. Finally, adding CLIPRemapper to our model highly boosts the manipulation performance, with slightly decreasing photorealism in terms of FID. Overall, these quantitative results demonstrate that the combination of the loss functions and the light-weight modules we use allows our model to perform well across all metrics and apply manipulations accurately while preserving the photorealism and preventing unwanted changes.

4.8 Limitations

As demonstrated by our experiments, our approach can successfully manipulate an input image with regards to a given target textual description. Our model utilizes pre-trained StyleGAN generator and inversion network, thus the the quality of the end results is heavily affected by the performance of these models. Apart from this limitation, which is common for the current state-of-the-start, the effectiveness of our method mainly lies in the proposed text-guided image encoder CLIPInverter, which estimates the residual latent code to capture the desired changes. Since CLIPInverter is trained by using a set of training images paired with corresponding textual descriptions, we observe that results of our approach might be affected by the biases that exist in the training data. For instance,

Fig. 12. **Qualitative results for the ablation study**. The global CLIP loss leads to unintuitive and unnatural results. Without perceptual losses, unwanted manipulations occur. Without the cycle pass or CLIPRemapper, we are not able to apply all the desired manipulations.

Fig. 13. **Limitations of our proposed CLIPInverter method**. Our approach might make some undesired changes to the given input image not mentioned in the provided textual description due to the biases that exist in the training set. This problem can be prevented by providing more comprehensive descriptions.

Multi-Modal CelebA-HQ dataset containing human face images consists of 10 descriptions for each image, but we observe that the descriptions are not diverse, often using similar adjectives referring to certain attributes. Moreover, there is an imbalance between the number of female and male images, causing a bias towards a specific gender in certain attributes. When only attributes are used in the textual descriptions without any pronouns, unexpected gender manipulations might occur due to these biases. As observed in Fig. 13, when we only use the description "*wavy hair*", a gender manipulation also occurs. We can alleviate this problem by using more comprehensive textual descriptions, including additional details such as "*She has wavy hair*", which yields a much more accurate manipulation. It is an interesting future direction to tackle the bias problem in a more systematic manner.

5 CONCLUSION

In this work, we have introduced CLIPInverter, a novel text-driven image editing approach. It can be used to manipulate an input image through the lens of StyleGAN latent space solely by providing a target textual description, which is much more intuitive than the commonly-used user inputs such as sketches, strokes or segmentation masks. The key component of our approach is the proposed text-guided adapter module called CLIPAdapter, which modulates image feature maps during the inversion to extract semantic edit directions with respect to the provided target description. Moreover, we suggest a text-guided refinement module that we refer to as CLIPRemapper, which performs an additional correction step on the predicted latent code from CLIPAdapter to further boost the accuracy of the done edits in the input image. Our model does not require an instance-level latent code optimization or a separate training for specific text prompts as done in the prior work, and thus provides a faster alternative to the approaches exist in the literature.

Our approach is not limited to a specific domain in that it only needs a pretrained StyleGAN model. As our experimental analysis on several different datasets illustrate, our model can handle the semantic edits through textual descriptions for very different domains. Moreover, thanks to the shared semantic space provided by the CLIP [Radford et al. 2021] model between images and text, our model can be also used to perform manipulations conditioned on another image or a novel textual description that has not been seen during training. Our experiments demonstrate significant improvements over the previous approaches in that our model can manipulate images with high accuracy and quality for any description.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4431–4440. https://doi.org/10.1109/ICCV.2019.00453

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Image2StyleGAN++: How to Edit the Embedded Images?. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. https://doi.org/10.1109/CVPR42600.2020.00832

Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. 2021. StyleFlow: Attribute-Conditioned Exploration of StyleGAN-Generated Images Using Conditional Continuous Normalizing Flows. *ACM Trans. Graph.* 40, 3, Article 21 (May 2021), 21 pages. https://doi.org/10.1145/3447648

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021a. Only a Matter of Style: Age Transformation Using a Style-Based Regression Model. *ACM Trans. Graph.* 40, 4, Article 45 (2021). https://doi.org/10.1145/3450626.3459805

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021b. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. 2021c. HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing. arXiv:2111.15666 [cs.CV]

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. http://arxiv.org/abs/1701.07875 cite arxiv:1701.07875.

Qingyan Bai, Yinghao Xu, Jiapeng Zhu, Weihao Xia, Yujiu Yang, and Yujun Shen. 2022. High-fidelity GAN inversion with padding space. In *European Conference on Computer Vision*. Springer, 36–53.

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. 2019a. Semantic Photo Manipulation with a Generative Image Prior. *ACM Trans. Graph.* 38, 4, Article 59 (jul 2019), 11 pages. https://doi.org/10.1145/3306346.3323023

David Bau, Jun-Yan Zhu Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. 2019b. Inverting Layers of a Large Generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1xsqj09Fm

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved Baselines with Momentum Contrastive Learning. arXiv:2003.04297 [cs.CV]

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. 2020. Editing in Style: Uncovering the Local Semantics of GANs. *CoRR* abs/2004.14367 (2020). arXiv:2004.14367 https://arxiv.org/abs/2004.14367

Antonia Creswell and Anil Anthony Bharath. 2016. Inverting The Generator Of A Generative Adversarial Network. *CoRR* abs/1611.05644 (2016). arXiv:1611.05644 http://arxiv.org/abs/1611.05644

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 4690–4699. https://doi.org/10.1109/CVPR.2019.00482

Prafulla Dhariwal and Alex Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. *CoRR* abs/2105.05233 (2021). arXiv:2105.05233 https://arxiv.org/abs/2105.05233

Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. 2022. HyperInverter: Improving StyleGAN Inversion via Hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Aditya Ramesh et al. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents.

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. 2021. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *CoRR* abs/2108.00946 (2021). arXiv:2108.00946 https://arxiv.org/abs/2108.00946

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 2672–2680.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2021. Vector Quantized Diffusion Model for Text-to-Image Synthesis. *arXiv preprint arXiv:2111.14822* (2021).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. *arXiv preprint arxiv:2006.11239* (2020).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. *CoRR* abs/1902.00751 (2019). arXiv:1902.00751 http://arxiv.org/abs/1902.00751

Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. 2022. Style Transformer for Image Inversion and Editing. *arXiv preprint arXiv:2203.07932* (2022).

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Drew A Hudson and C. Lawrence Zitnick. 2021. Generative Adversarial Transformers. *Proceedings of the 38th International Conference on Machine Learning, ICML 2021* (2021).

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Hk99zCeAb

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks.. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-Based Real Image Editing with Diffusion Models. https://doi.org/10.48550/ARXIV.2210.09276

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. 2022. DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2426–2435.

Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. 2021. Exploiting Spatial Dimensions of Latent in GAN for Real-time Image Editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar Yanardag. 2021. StyleMC: Multi-Channel Based Fast Text-Guided Image Generation and Manipulation. In *WACV*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. 2020. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. 2020. Manigan: Text-guided image manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7880–7889.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Takeru Miyato and Masanori Koyama. 2018. cGANs with Projection Discriminator. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ByS1VpgRZ

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *CoRR* abs/2112.10741 (2021). arXiv:2112.10741 https://arxiv.org/abs/2112.10741

Weili Nie, Arash Vahdat, and Anima Anandkumar. 2021. Controllable and compositional generation with latent-space energy-based models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. 2022. Spatially-Adaptive Multilayer Selection for GAN Inversion and Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2085–2094.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *CoRR* abs/1705.08045 (2017). arXiv:1705.08045 http://arxiv.org/abs/1705.08045

Sylvestre-Alvise Rebuffi, Andrea Vedaldi, and Hakan Bilen. 2018. Efficient Parametrization of Multi-domain Deep Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8119–8127. https://doi.org/10.1109/CVPR.2018.00847

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *ACM Trans. Graph.* (2021).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. https://doi.org/10.48550/ARXIV.2205.11487

Maximilian Seitzer. 2020. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid. Version 0.2.1.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yujun Shen and Bolei Zhou. 2021. Closed-Form Factorization of Latent Semantics in GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ayush Tewari, Mohamed Elgharib, Mallikarjun B. R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020a. PIE: Portrait Image Embedding for Semantic Control. *CoRR* abs/2009.09485 (2020). arXiv:2009.09485 https://arxiv.org/abs/2009.09485

Ayush Tewari, Mohamed Elgharib, Mallikarjun B R, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020b. PIE: Portrait Image Embedding for Semantic Control. *ACM Trans. Graph.* 39, 6, Article 223 (nov 2020), 14 pages. https://doi.org/10.1145/3414685.3417803

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. *ACM Trans. Graph.* 40, 4, Article 133 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459838

Dani Valevski, Matan Kalman, Y. Matias, and Yaniv Leviathan. 2022. UniTune: Text-Driven Image Editing by Fine Tuning an Image Generation Model on a Single Image. *ArXiv* abs/2210.09477 (2022).

Andrey Voynov and Artem Babenko. 2020. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*. PMLR, 9786–9796.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *The Caltech-UCSD Birds-200-2011 Dataset.* Technical Report CNS-TR-2011-001. California Institute of Technology.

Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. 2022. High-Fidelity GAN Inversion for Image Attribute Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. 2022. Hairclip: Design your hair by text and reference image. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022).

Zongze Wu, Dani Lischinski, and Eli Shechtman. 2020. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. *CoRR* abs/2011.12799 (2020). arXiv:2011.12799 https://arxiv.org/abs/2011.12799

Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021a. TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. 2021b. GAN Inversion: A Survey. *arXiv preprint arXiv: 2101.05278* (2021).

Mohit Bansal Yi-Lin Sung, Jaemin Cho. 2022. VL-Adapter: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. In *CVPR*.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. 2021. StyleSwin: Transformer-based GAN for High-resolution Image Generation. arXiv:2112.10762 [cs.CV]

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE Computer Society, 586–595. https://doi.org/10.1109/CVPR.2018.00068

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain GAN Inversion for Real Image Editing. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. 2016. Generative Visual Manipulation on the Natural Image Manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.