CoralStyleCLIP: Co-optimized Region and Layer Selection for Image Editing

Ambareesh Revanur* Debraj Basu Shradha Agrawal Dhwanit Agarwal Deepak Pai
Adobe
*arevanur@adobe.com

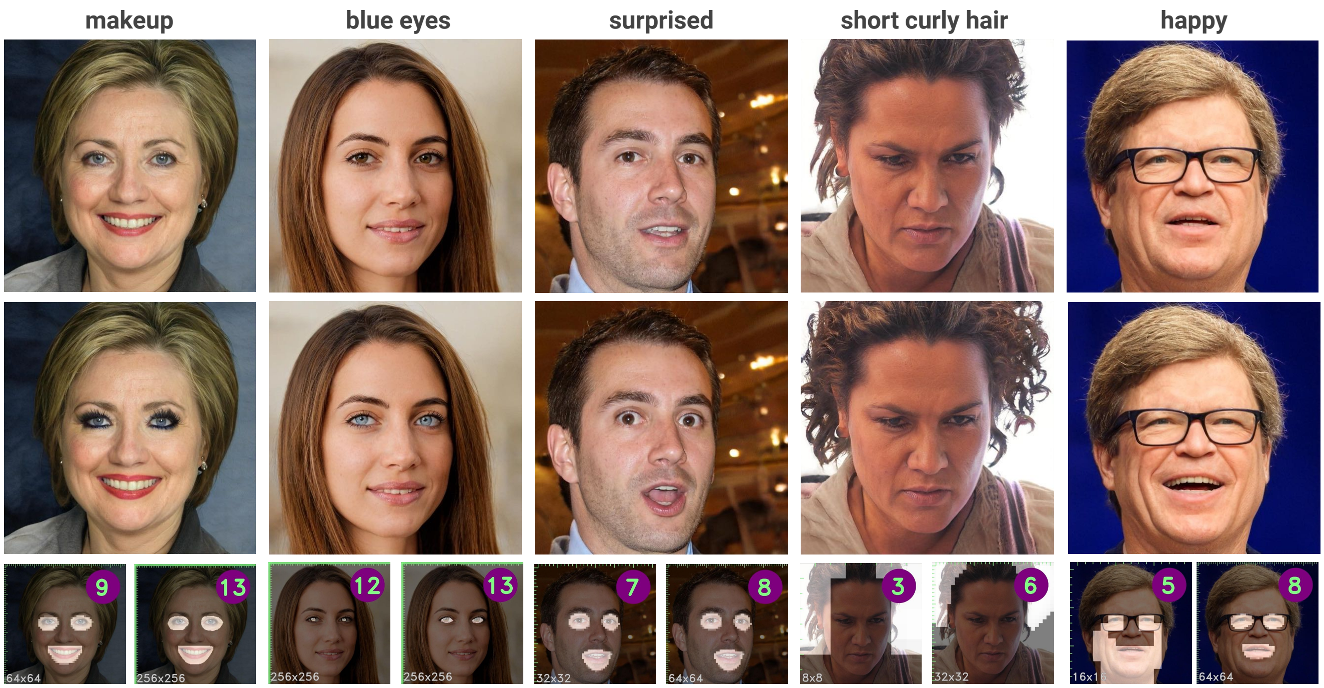

Figure 1 The original and edited images using CoralStyleCLIP are shown in the first and second rows of images, respectively. The bottom row shows the regions and StyleGAN2 layer numbers automatically selected for editing. The driving text prompts are above every column.

Abstract

Edit fidelity is a significant issue in open-world controllable generative image editing. Recently, CLIP-based approaches have traded off simplicity to alleviate these problems by introducing spatial attention in a handpicked layer of a StyleGAN. In this paper, we propose CoralStyleCLIP, which incorporates a multi-layer attention-guided blending strategy in the feature space of StyleGAN2 for obtaining high-fidelity edits. We propose multiple forms of our co-optimized region and layer selection strategy to demonstrate the variation of time complexity with the quality of edits over different architectural intricacies while preserving simplicity. We conduct extensive experimental analysis and benchmark our method against state-of-the-art CLIP-based methods. Our findings suggest that CoralStyleCLIP results in high-quality edits while preserving the ease of use.

1. Introduction

Controlling smooth semantic edits to photorealistic images [1, 5, 34, 41] synthesized by well-known Generative Adversarial Networks (GANs) [13, 18, 19] has become simplified with guidance from independently trained contrastive models such as CLIP [36]. Using natural language as a rich medium of instruction for open-world image synthesis [39, 46–48] and editing [11, 12, 24, 26, 28, 45] has addressed many drawbacks of previously proposed methods.

As first demonstrated by StyleCLIP [34], the requirements for large amounts of annotated data [25] and manual efforts [14, 44] were considerably alleviated. Furthermore, the range of possible edits that were achievable significantly improved [34]. The underlying theme of related approaches involves CLIP-driven exploration [15, 22, 34] of the intermediate disentangled *latent spaces* of the GANs.

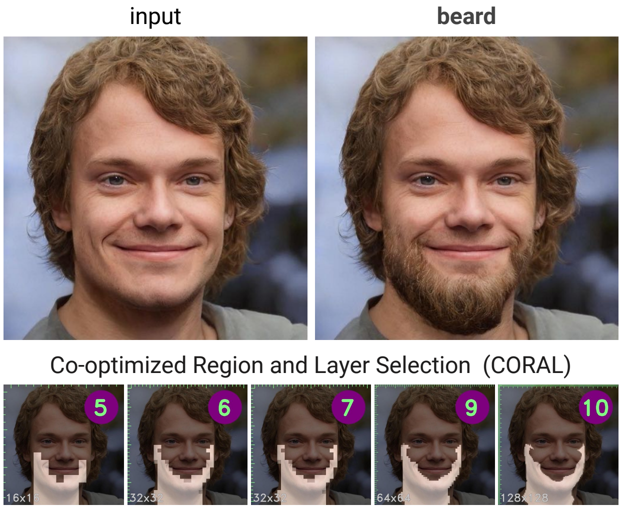

Figure 2 For achieving *beard*, CORAL selects appropriate regions in layers 5-10 for carrying out the required coarse edits in early layers and finer texture edits in later layers

It is well understood by now that manipulating the latent code of a StyleGAN for aligning with a *text prompt* can be computationally intense, as seen in StyleCLIP latent optimization [34], as well as the latent mapper methods [34]. This presents a trade-off between the complexity and quality of edits leveraged by StyleCLIP global directions [34] and StyleMC [22].

In addition, these methods often result in undesirable edits to unexpected regions of an image (see [15]), addressed to some extent by FEAT [15]. However, FEAT requires manual intervention, as described in Section 2, and involves significant training complexity of the order of hours[1].

Contributions. In this paper, we propose CoralStyleCLIP, which addresses these challenges by combining the ease of use [34] with efficient [22] high fidelity edits [15] into our approach. In particular, we propose a novel strategy which, for a given text prompt, jointly learns both the appropriate direction of traversal in the latent space, as well as which spatial regions to edit in every layer of the StyleGAN2 [19] (see Figure 1, Figure 2) without any mediation.

Our approach overcomes the need for manual effort in selecting an appropriate layer for FEAT by incorporating multi-layer feature blending to enable the joint learning process. As a result, the edits are very accurate, rendering our method simple and effective.

The co-optimized regions and layers jointly learned with appropriate latent edits typically select earlier layers for enacting coarse edits, such as shape and structural, compared to finer edits, such as color and texture, which are usually orchestrated through the latter layers of the StyleGAN2.

To alleviate the time complexity, we implement this strategy for *segment selection* (see Section 3.2), where we jointly learn a *global direction* [22,34] in the $\mathcal{W}^+$ space and limit the predicted areas of interest at every layer to segments from a pre-trained segmentation network. Doing so reduces the learning complexity significantly (see Table 1), albeit with potential pitfalls discussed in Section 4.3. We mitigate these pitfalls with a jointly trained *attention network* where we relax the areas of interest at every layer to spatial masks predicted by the network (see Section 3.2). As a result, the training time increases from a few minutes to about an hour while improving the quality of edits compared to the *segment selection* approach.

In summary, our contributions are as follows:

- We propose a novel multi-layer blending strategy that attends to features selectively at the appropriate StyleGAN layer with minimal hand-holding.
- A CORAL variant based on *segment selection* demonstrates high edit quality at a fraction of time cost.
- Through extensive empirical analysis, we find that CORAL outperforms recent state-of-the-art methods and is better equipped to handle complex text prompts.

2. Related Work

The use of generative models for high-quality image synthesis and manipulation has a rich history [7, 8, 17]. In particular, the disentangled latent spaces of StyleGAN provide robust interpretable controls for editing valuable semantic attributes of an image [3,9,14,30,41–44]. Desirable changes to attributes of interest were previously brought out by discovering the relevant channels [44] and curating principal components [14] either through manual inspection or otherwise driven by data-hungry attribute predictors.

StyleFlow [4] leverages normalizing flows to perform conditional exploration of a pre-trained StyleGAN for attribute-conditioned image sampling and editing. By learning to encode the rich local semantics of images into multi-dimensional latent spaces with spatial dimension, StyleMapGAN [20] demonstrates improved inversion quality and the benefits of spatially aware latent code interpolation between source and target images for editing purposes. The advent of CLIP [36] has re-ignited interest in open domain attribute conditioned synthesis of images [34, 35, 50]. Text-driven edits have considerably reduced both the time and effort required for editing images and extended the range of possible edits significantly [34], all the more with increased interest in diffusion models [29, 37, 38].

The disentangled nature of the latent spaces of StyleGAN has facilitated heuristics such as a fixed global direction in StyleCLIP [34] and, more recently, StyleMC [22]. For training efficiency, StyleMC performs CLIP-driven optimization on the image generated at a low-resolution layer

[1] With no official implementation available, we present comparisons with our reimplementation of FEAT denoted by FEAT* in this paper.

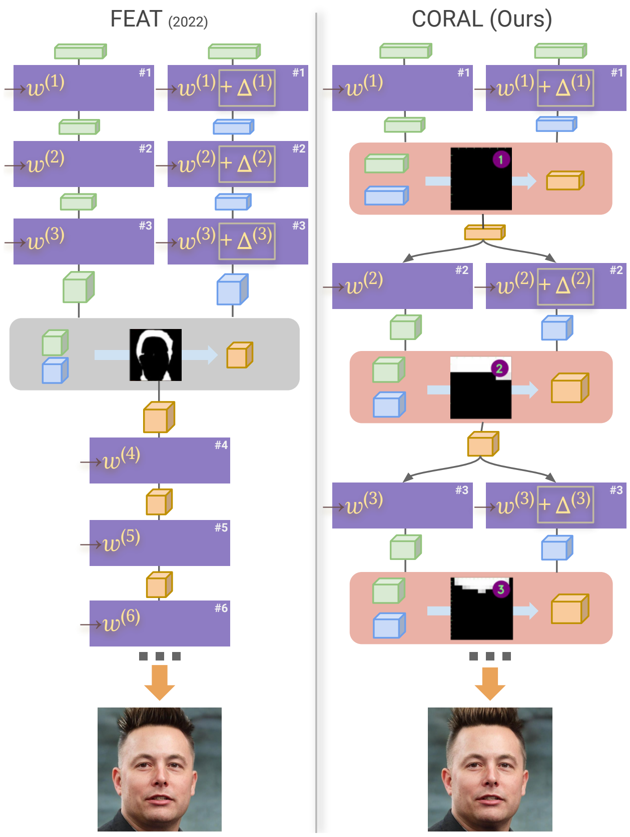

Figure 3 Comparison of FEAT [15] with CORAL. In FEAT (left), the spatial features are blended at a carefully hand-picked layer l. CORAL (right) performs multi-layer blending with custom edit regions per layer.

of the StyleGAN. Unfortunately, this limits the range of edits to only those possible by manipulating latent codes at the earlier layers.

For ameliorating edits in unexpected regions of an image, strategies for blending latent features have been an emerging theme in many recent papers [6, 15, 16, 20]. [6, 15, 20] interpolate spatial features more explicitly. In contrast, StyleFusion [16] realizes similar objectives through blended latent code extracted using a fusion network that combines disjoint semantic attributes from multiple images into a single photorealistic image.

Our work is most closely related to [15, 22, 34]. FEAT [15] reduces undesirable edits by imposing sparsity in the number of spatial features modified by StyleCLIP at a manually selected layer l of the StyleGAN2. FEAT edits layers $\leq l$ using a non-linear latent mapper, while the attention network emits a spatial mask for interpolating edited spatial features at layer l with original spatial features at the same layer (see Figure 3). At the cost of training time and convenience, FEAT achieves high-fidelity edits. If the blending layer is not carefully selected, the edits can be significantly poor, as shown in [15] and Figure 6. Furthermore, FEAT enacts inferior edits when presented with multi-faceted prompts (see Figure 7). In Suppl., we also discuss how CORAL is different from a multi-layer extension of FEAT.

Furthermore, we argue that the required edits for aligning with a given text prompt arise from multiple layers of the StyleGAN2, necessitating a multi-layer feature interpolation mechanism (see Section 3.3). Our method percolates meaningful edits from the current layer onto subsequent layers, with restrictions on the number of spatial edits customized for each layer. As a result, we can automatically select the correct layers and regions for editing an image.

To correctly identify the region of interest at every layer, we discuss a lightweight segment-selection scheme (see Section 3.2) and contrast this with an involved convolutional mask prediction model motivated by FEAT. Recently, SAM [32] accomplished superior GAN inversion at the cost of editability by leveraging different latent spaces of the StyleGAN2 in a spatially adaptive manner. However, the edits performed on the inverted latent codes continue to modify irrelevant image regions and could benefit from CORAL (see Section 3).

With a focus on convenience and fidelity, CoralStyleCLIP learns global directions at every layer of the StyleGAN2, as done in [22], and exhibits high-quality edits with a significant reduction in the training time and manual effort (see Table 1). Borrowing inspiration from [34], we also implement our co-optimized region and layer selection strategies for a non-linear mapper-based latent edit and demonstrate additional customized and high fidelity edits.

3. Approach

An image edit is often spatially localized to a specific region of interest. For example, edits corresponding to the *mohawk* text prompt should affect only the hair region of the portrait image while preventing edits in other parts. In this work, we learn a latent edit vector and a soft binary mask at every layer of a StyleGAN2 to accurately edit the image according to the input text prompt. We achieve this by training them end-to-end while respecting the challenging but desirable minimal overall edit area constraint. Following a brief revisit to the StyleGAN architecture, we introduce two simple yet effective strategies to determine the region of interest given a text prompt. Finally, we introduce a novel multi-layer blending strategy that is vital for achieving high fidelity minimal edits.

3.1. Background

StyleGAN2 [19] is a state-of-the-art model trained for generating high-resolution images typically of sizes 1024×1024 or 512×512. The network consists of a mapper module that maps a random vector $z \in \mathcal{Z} \sim \mathcal{N}(0, 1)$ to a vector in $w \in \mathcal{W}$ space via a multi-layer perceptron (MLP), and a generator module comprising 18 convolutional blocks.

The $\mathcal{W}^+$ space, first defined by [2], is a concatenation of 18 different $w^{(l)}$ vectors where $l \in \{1, 2, \ldots, 18\}$. The $w^{(l)}$ instance in $\mathcal{W}^+$-space is first transformed through a layer-specific affine operation to obtain *stylecode* $s^{(l)} \in \mathcal{S}$, at all layers of the generator module. The input to the generator module is a learned tensor of 4×4 resolution. It is gradually increased to a resolution of 1024×1024 as the input tensor is passed down through the layers of the generator.

We denote the constant input tensor as c and the feature obtained at a layer l as $f^{(l)}$. Further, we denote the $\mathcal{W}^+$ code at layer l as the $w^{(l)}$ and a layer in generator module as $\Phi^{(l)}$. Therefore, $f^{(l)}$ can be expressed as $f^{(l)} = \Phi^{(l)}(f^{(l-1)}, w^{(l)})$, where $l \in \{1, 2, \cdots, 18\}$, $c = f^{(0)}$ and the generated image $I = \sum_{l=1}^{18} RGB^{(l)}(f^{(l)})$.

In our work, we aim to find a latent vector $\Delta^{(l)}$ in the $\mathcal{W}^+$ such that the image generated by the latent code $w^{(l)} + \Delta^{(l)}$ applied to every layer of generator results in an edited image I^*. For simplicity, we denote f^* and $w^* = w + \Delta$ as edited features and $\mathcal{W}^+$ latent code, respectively. Therefore, we have $f^{*(l)} = \Phi^{(l)}(f^{*(l-1)}, w^{(l)} + \Delta^{(l)})$. A recent study showed that StyleGAN2 learns global attributes such as position in earlier layers, structural changes in middle layers, and appearance changes (e.g., color) in the final set of layers [18, 45]. However, determining the right set of layers for a given text prompt is challenging and has been explored only empirically in FEAT [15].

3.2. Co-optimized region and layer selection (CORAL)

We aim to edit the image to match the text prompt with minimal changes. To this end, the first step is correctly identifying the region of interest. Further, given the diversity and richness of latent space at each layer in the generator, we posit that the edits to the image can come from multiple layers of the StyleGAN2 generator.

To address both requirements, we introduce CORAL, a co-optimized region and layer selection mechanism. In CORAL, we propose two simple-yet-effective approaches for learning a soft binary mask $m^{(l)} \in [0, 1]^{f^{(l)}_{dim}}$ at every layer of the generator module with the same height and width dimensions as the feature resolution at the given layer.

CORAL based on segment-selection. We can use any off-the-shelf pre-trained semantic segmentation network to determine the region of interest in this approach. Intuitively, existing image segmentation networks generally capture semantic parts of the image that we are interested in editing, such as eyes, mouth, and lips. Therefore in many cases, this problem can be posed as selecting the appropriate segments. To achieve this, we introduce a matrix e of dimension $P \times 18$ where P is the number of classes predicted by the segmentation network. Each entry in the matrix e is in the range $[0, 1]$, where 1 represents a confident segment selection for the given text prompt t.

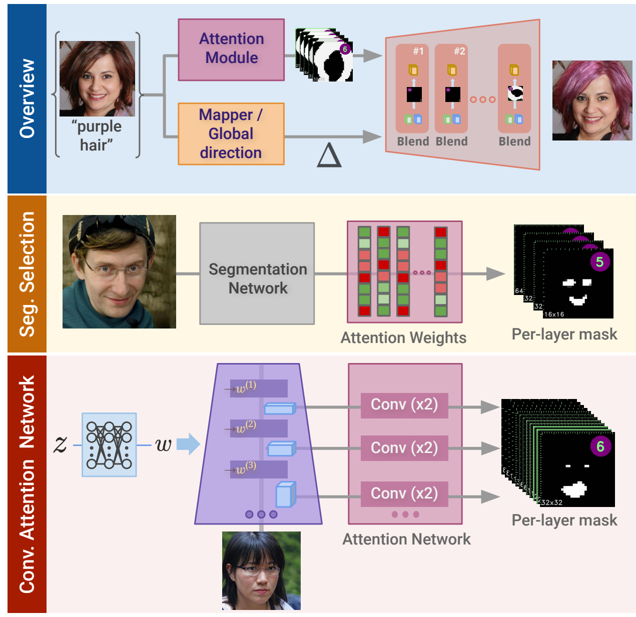

Figure 4 Overview of CoralStyleCLIP. The only trainable components are the attention module and the mapper/global direction. Two different variants of the attention module are summarized as segment selection and convolutional attention network (see Section 3.2 for more details).

The matrix e is converted into a spatial mask $m^{(l)}$ by masking the segments with the confidence values and resizing the segmentation map to the resolution of the feature maps at each layer. In the training phase, the parameters in the matrix e are trained after applying a sigmoid, and during inference, we apply a prompt-specific threshold τ_t to the sigmoid. As depicted in Figure 4, the only trainable parameters in this pipeline are e. Therefore, this can achieve desirable edits with high accuracy up to 8x faster than FEAT [15].

CORAL based on convolutional attention network. Segment-selection-based CORAL is limited by the segments available in the pre-trained network. As shown in results Figure 6-F, the segment-selection method is prone to over-selection or under-selection of the region of interest. To overcome this limitation, we implement an attention network that directly predicts the masks $m^{(l)}$ at every layer of the generator as shown in Figure 4. In this architecture, we obtain a mask with the exact resolution as that of the corresponding feature in the layer. Unlike FEAT, we hypothesize that the mask at a layer l should depend only on the features $f^{(l)}$ available at the current layer since we are interested in predicting the mask at every layer.

Despite incurring higher training costs from having to learn the convolutional layers, the masks produced with this approach are smoother and avoid over/under-selection issues by accurately predicting the correct region of interest.

3.3. Multi-layer feedforwarded feature blending

CORAL produces soft binary masks $m^{(l)}$ at every layer of the generator module. These masks blend features such that the features corresponding to the confident regions are borrowed from features f^* generated with updated style code, and on similar lines, features from non-confident regions are borrowed from original features f of the unedited image. This ensures that we only modify the regions corresponding to the text prompt and prevents modifications of non-masked regions. Unfortunately, a 0-mask (completely black mask) at any layer would throw away any updated feature information from the previous layers and would propagate the original features f from that point onward.

To prevent this bottleneck, we design a novel multi-layer feature blending strategy (see Figure 3) that utilizes a parallel pathway where the feature obtained from layer $l-1$ is passed through the generated block Φ twice - once with the original latent code w and another pathway with updated latent code $w+\Delta$ to obtain two feature sets for blending. The former feature can be viewed as a feature that is not edited but has all the information propagated from previous layers. The multi-layer blending strategy expressed in (1), ensures that no feature information is lost along the way.

$$\begin{aligned} \widehat{f^{*(l)}} &= \Phi^{(l)}(f^{*(l-1)}, w^{(l)} + \Delta^{(l)}) \\ \widehat{f^{(l)}} &= \Phi^{(l)}(f^{*(l-1)}, w^{(l)}) \\ f^{*(l)} &= m^{(l)} \odot \widehat{f^{*(l)}} + (1 - m^{(l)}) \odot \widehat{f^{(l)}} \end{aligned} \tag{1}$$

Intuitively, when the mask is completely blank (which is often desirable to keep the edits to a minimum), the features are feedforwarded simply with edits from previous layers.

3.4. Types of latent edits

For a given convolutional layer l, when the learned latent edit $\|\Delta^{(l)}\| > 0$, the corresponding feature $\widehat{f^{*(l)}}$ in (1) incorporates attributes which are desirable for semantic alignment with the given text prompt. The mask $m^{(l)}$ counteracts possible undesirable artifacts through a region-of-interest-aware interpolation strategy.

The $\Delta^{(l)}$ by itself is, however, well studied in [22, 34], both of which identify a single global direction that can semantically edit images for a given text prompt. Such a simple parameterization does result in accurate edits for simple text prompts, as discussed in [34].

Our findings suggest that training time is significantly reduced for prompts where a global direction can affect desirable changes. However, a more involved image-dependent non-linear mapper model $g(\cdot)$ as a function of $w^{(l)}$ at every layer can affect such changes with higher precision.

Therefore, we implemented CORAL for both versions of latent edits: **(i)** *global direction*; **(ii)** *latent mapper*. The latent mapper $g(\cdot)$ is an MLP-based model along the lines of [34, Section 5], where the $w^{(l)}$ are split into three groups: coarse (l in 1 to 4), medium (l in 5 to 8) and fine (l in 9 to 18); and each of these groups is processed by a different MLP[2]. Our multi-layer feature blending mechanism is independent of the parametrization of the latent edit, which is jointly learned with the mask $m^{(l)}$ predictors.

3.5. Loss formulation

We now describe our proposed methods' training strategy and loss formulation. We are given a text prompt t and an image with corresponding $\mathcal{W}^+$ code w. The goal is to find the right region of interest using a CORAL framework and determine the latent vector to help with the image edit. The only trainable components in our approach are the latent vector Δ and the parameters in the CORAL framework. In the case of segment selection, the only trainable component in CORAL is the matrix, and in the case of convolutional attention networks, the Conv layers in the attention network are trainable.

CLIP loss: The first key loss component is the CLIP loss originally proposed in StyleCLIP [34]. The idea is to use the pre-trained CLIP model to edit the latent vector such that the embedding of the image I^* produced aligns with the embedding of the text prompt t in CLIP's latent space.

In addition we also synthesize the image $\widetilde{I}$, by setting $m_{i,j}^{(l)} = 1 \ \forall i, j, l$ in (1) and compute its CLIP loss. To understand this, we can envision I^* as a sophisticated non-linear interpolation between I_0 and $\widetilde{I}$ using strategies given in (1). Here I_0 is the original unedited image.

By simply imposing a CLIP loss on I^*, $\widetilde{I}$ remains unrestricted and can potentially contain undesirable artifacts, as long as I^* aligns with the text prompt t. However, our region selectors in Section 3.2 derive their supervision from $\widetilde{I}$ and might also learn to include these artifacts. Our final semantic alignment loss is as follows:

$$\mathcal{L}_{clip} = \frac{1}{2}\left(D_{\text{CLIP}}(I^*, t) + D_{\text{CLIP}}(\widetilde{I}, t)\right) \tag{2}$$

L_2 loss: Controlled perturbations to the latent spaces of a StyleGAN2 can result in smooth semantic changes to the generated image. As a result, we optimize the squared Euclidean norm of Δ, i.e., $\mathcal{L}_{l_2} = \|\Delta\|_2^2$, in the $\mathcal{W}^+$ space to prefer latent edits with smaller l_2 norms.

ID loss: In order to prevent changes to the identity of a person during image manipulation, we impose an ID loss $\mathcal{L}_{id} = 1 - \langle \mathcal{R}(I^*), \mathcal{R}(I) \rangle$ using cosine similarity between the embeddings in the latent space of a pre-trained ArcFace network $\mathcal{R}$ [10, 22, 34, 40].

Minimal edit-area constraint: We encourage the network to find an edit with changes to compact image areas. In the case of segment selection, this is achieved by penalizing the CORAL matrix e as follows:

[2]Unlike in [34], we remove the `LeakyReLU` activation after the final fully connected layer, as it empirically expedites the optimization.

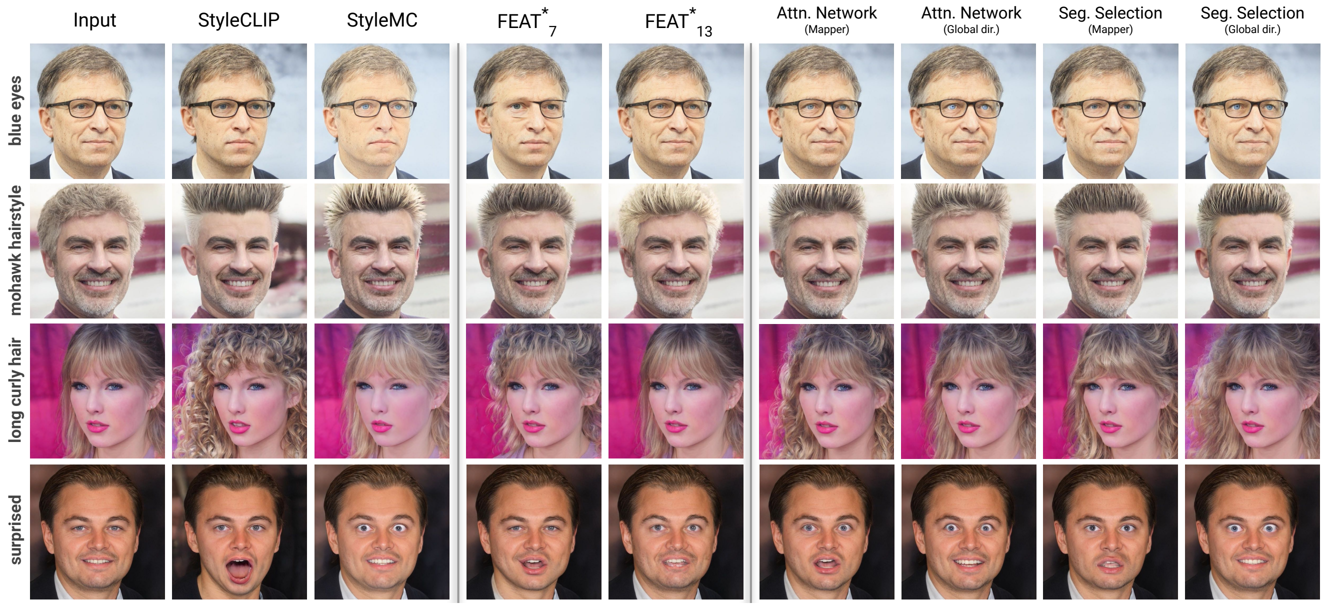

Figure 5 Comparison of variants of CORAL differing in complexity with closely related FEAT* [15], StyleCLIP mapper method [34] and StyleMC [22]

$$\mathcal{L}_{area}^{ss} = \sum_{i,j} e_{i,j} \tag{3}$$

In the case of a convolutional attention network, this is achieved by imposing the minimal edit constraint directly on the masks m as follows:

$$\mathcal{L}_{area}^{can} = \sum_{l} n_l \Big(\sum_{i,j} m_{i,j}^{(l)} \Big) \tag{4}$$

where n_l is a normalizing constant defined per layer to account for the growing feature dimensions as the feature passes through the StyleGAN2 generator module.

Smoothness loss: In the case of the convolutional attention network, it would be desirable to predict a smooth mask. This is achieved by imposing a total variation loss [15].

$$\mathcal{L}_{tv} = \sum_{i,j,l} \|m_{i,j}^{(l)} - m_{i+1,j}^{(l)}\|_2^2 + \sum_{i,j,l} \|m_{i,j}^{(l)} - m_{i,j+1}^{(l)}\|_2^2 \tag{5}$$

In summary, the loss formulations for the segment selection and convolutional attention mechanisms are as follows:

$$\mathcal{L}_{ss} = \mathcal{L}_{clip} + \lambda_{l_2}\mathcal{L}_{l_2} + \lambda_{id}\mathcal{L}_{id} + \lambda_{area}\mathcal{L}_{area}^{ss} \tag{6}$$

$$\mathcal{L}_{can} = \mathcal{L}_{clip} + \lambda_{l_2}\mathcal{L}_{l_2} + \lambda_{id}\mathcal{L}_{id} + \lambda_{area}\mathcal{L}_{area}^{can} + \lambda_{tv}\mathcal{L}_{tv} \tag{7}$$

Both the CORAL module and the latent editor are optimized in an end-to-end fashion using the above losses.

4. Experiments

We evaluate CORAL mainly in the context of human faces and demonstrate high-quality edits to photo-realistic faces of size 1024×1024 generated by a StyleGAN2 pre-trained on the FFHQ dataset [18]. We present additional results on *sketch* and *pixar* domains as well as Cars dataset [23] in Suppl. For both variants of CORAL in Section 3.2, we compare edits from the *global direction* and *latent mapper* in Section 3.4. All hyperparameter configurations for (6) and (7) are provided in Suppl.

4.1. Training and inference

The loss functions corresponding to the two different variants of CORAL are given by (6) and (7). Our experiments were conducted on one NVIDIA Tesla P40 24 GB GPU with a batch size of 3. The latent editor and CORAL modules are jointly optimized using an Adam optimizer [21] while keeping the StyleGAN2 fixed.

For a given text prompt t, a data point is given by a randomly sampled standard normal vector $z \sim \mathcal{Z}$ space, and the maximum number of iterations is set to 20,000. However, in Table 1, we note that the training time required for achieving the desired quality of edit increases as we switch from segment-selection to a convolutional attention network, the same as in going from global direction edits to training a latent mapper.

Furthermore, during inference, we limit the automatically selected regions for editing by setting $m^{(l)} \leftarrow m^{(l)} \odot \mathbf{1}\{m^{(l)} \geq \tau\}$ where τ is typically 0.85. For applying desirable edits and reversing them (see Figure 6-G), we have a multiplying factor $\alpha \in [-1.5, 1.5]$ for the edit direction Δ.

Out of the 18 convolutional blocks and the corresponding w code per layer, our CORAL strategy and the latent edits, as well as edits from our baselines, are only performed on the first 13 layers, which are known to span coarse and fine controls over diverse attributes [44] such as expressions, age, style and color of facial hair, and eyes, among others.

Segment selection: Based on ideas from [31], a pre-trained mixture model is used for performing unsupervised semantic segmentation of the StyleGAN2 generated images into 5 classes per pixel. This model is then used to determine the region of interest with CORAL based on segment selection. In Figure 6-E, we also compare with CORAL for a weakly

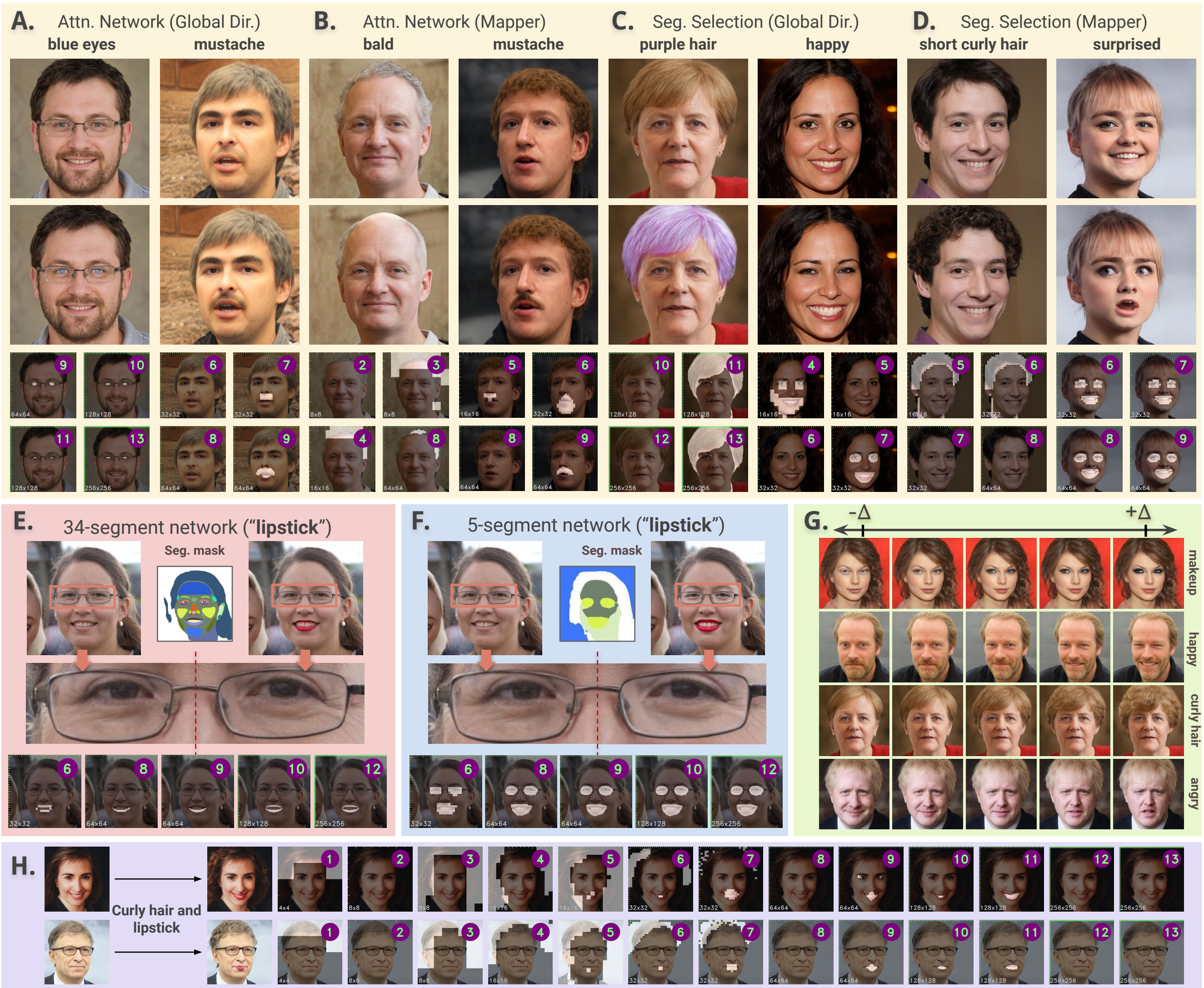

Figure 6 Each column in figures A to D demonstrates a text-driven edit on an input image along with the corresponding layers and regions selected. As a limitation of segment selection, we observe over-selection of the region of edit in figure F, which is absent in E. Figure G compares edits along both the positive and negative direction where we observe intuitive differences between removal and application of *makeup*, *happy* vs. *unhappy* and *curly* vs. *smooth hair*. Finally, Figure H demonstrates the edit regions selected by CORAL across different layers of the StyleGAN2 for a complex prompt.

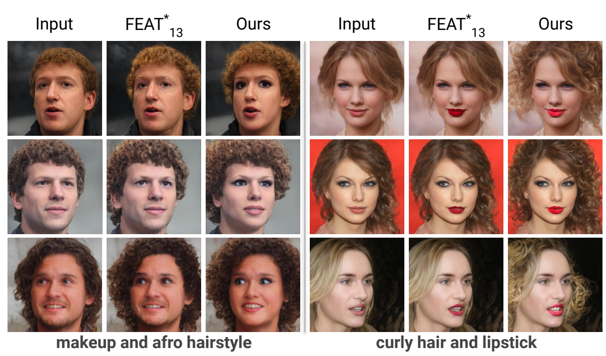

Figure 7 Comparison of CORAL with FEAT$^*_{13}$ for multi-faceted prompts.

supervised 34 class DatasetGAN [49] network, trained on the features of StyleGAN2 network using few shot labels.

Attention network: At each convolutional block $l \in [1, 2, \dots, 13]$ of the StyleGAN2, the attention network first applies 32 different 1×1 convolutional filters upon the spatial features $f^{(l)}$ to reduce the number of channels to 32 followed by ReLU [27] activation, after which another 1×1 convolutional layer and sigmoid activation are applied to obtain $m^{(l)}$. We set $n_l = {}^1\!/_{\mathrm{size}[l]}$ in (4), where size$[l]$ is given based on the height and width of $f^{(l)}$, for example if the resolution is 32×32, then $n_l = {}^1\!/_{32}$.

4.2. Evaluation

Our method is most closely related to StyleCLIP [34], StyleMC [22] and FEAT [15]. For a comparison with CORAL, we run the official implementation of the latent mapper technique of StyleCLIP, as well as a re-

Table 1 Average Clean-FID [33] and training time to desirable quality. Text-prompt legend: **T1)** Happy; **T2)** Surprised; **T3)** Blue eyes; **T4)** Mohawk hairstyle. Method legend: **1)** SS Global; **2)** SS Mapper; **3)** AttnNet-Global; **4)** AttnNet-Mapper; **5)** StyleCLIP; **6)** StyleMC; **7)** FEAT*

		Clean-FID (↓)					Avg. Time
		T1	T2	T3	T4	Avg.	
CORAL	1	4.23	4.42	6.73	6.05	5.35	15min
	2	3.75	9.28	1.75	2.73	4.38	42min
	3	2.22	2.26	3.08	7.20	3.69	1.2hr
	4	1.85	6.38	1.38	5.29	3.73	2hr
others	5	6.98	18.14	5.40	22.50	13.26	1.5hr
	6	2.93	25.12	22.15	11.30	15.38	20s
	7	2.51	9.46	1.93	3.01	4.23	1.8hr

implementation of StyleMC[3] which optimizes a single global direction across multiple images, only for layers of the StyleGAN2 until resolution 256×256 (see [22]).

Without an official implementation of FEAT, we evaluate our re-implementation of FEAT denoted by FEAT* with $l \in \{7, 13\}$. We maintain equivalent settings in the design of the latent mapper and attention network and only intend to compare the single-layer FEAT-style blending with our multi-layer feedforwarded blending (see Figure 3).

4.3. Results

Merits: In Figure 5, we observe that both the StyleCLIP mapper method and StyleMC result in undesirable edits, such as irrelevant edits to the background. StyleCLIP also reduces the age in the first row and affects the neck region. In the fourth row, we see that in addition to applying the prompt *surprised*, it discards the white shirt. StyleMC affects the first three rows' complexion, facial expression, and hair color. As also seen in [15], we find that for finer edits (row 1), FEAT-style blending at layer 13 (FEAT^*_{13}) is preferable as also with FEAT^*_{7} for coarse edits (rows 2-4). We find that *blue eyes* results in unwanted edits when blended at $l = 7$, and so does *mohawk hairstyle* at $l = 13$.

CORAL, however (last four columns in Figure 5), only affects the relevant regions of interest, which would be the hair region for *long curly hair* and *mohawk hairstyle*, the eyes and mouth for *blue eyes* as well as *surprised*. These traits persist in Figure 1, and Figure 6-A to D wherein the edits are incorporated such that the editing area is minimal and is limited to only the relevant layers. CORAL learns the layers and regions to edit automatically with no domain knowledge or repeated trials. The edits are highly accurate. For example, the prompt *mustache* does not also affect the beard, as is apparent from the corresponding masks.

Under the minimality constraints given by (3) and (4), we observe that for enabling finer edits such as *blue eyes* and *purple hair*, only the latter higher resolution layers (typically $l \geq 8$) are selected, whereas, for coarser structural edits, the earlier smaller layers (typically $l \leq 8$) are automatically selected. We clearly see that when CORAL is trained for complex multi-faceted prompts such as *curly hair and lipstick* (see Figure 7 and Figure 6-H), the hair edits come from earlier layers whereas the lip edits come from last layers. Furthermore, for such prompts, we found that FEAT blending fails to preserve *realism* by introducing noise artifacts (see the example for FEAT^*_{13} under *makeup and afro hairstyle* in Figure 7). This is also seen in Figure 5 for *mohawk* using FEAT^*_{13}.

From Table 1, we see that while the Clean-FID [33] of all our edits remains within acceptable bounds of the initially generated distribution, the time required to train CORAL to a desirable edit quality increases with the complexity of the region, layer selector, and the latent editor combined, from method 1 to 4. Segment-selection-based CORAL is significantly faster to train than the attention network.

We also observe that, on average, segment selection has a higher FID than attention network. Along similar lines, *global* edits have a higher FID than *latent mapper*, except for *surprised*, which we attribute to *global* edits predominantly affecting the eyes for this prompt, even for StyleMC, unlike the mapper method which also opens up the mouth.
Limitations: The segment-selection-based approach trains at a fraction of the time taken by its counterparts, as seen in Table 1. However, the defined segments of a pre-trained segmentation model can affect performance. For example, in Figure 6-F, our semantic segmentation model combines all the eye and mouth regions into a single semantic segment. As a result, the text prompt *lipstick* brightens the skin tone and removes wrinkles from around the eyes. Alternatively, in Figure 6-E, a different segmentation network with dedicated classes for lips overcomes this issue.

We also note that the quality of the mustache is superior in Figure 6-B compared to A. It turns out that unlike our non-linear mapper which succeeds, the *global* edits result in black coloration in the *mustache* region in many examples.
Ethical aspects: In line with current works, we benchmark our approach using publicly available celebrity images [34]. Although our approach demonstrates superior edits on diverse faces, our approach still inherits biases present in StyleGAN and CLIP models. Further, a generative model (e.g., CORAL) could be misused to create fake information.

5. Conclusion

CoralStyleCLIP leverages StyleGAN2 and CLIP models to co-optimize region and layer selection for performing high-fidelity text-driven edits on photo-realistic generated images. We demonstrate the efficacy of our generic multi-layer feature blending strategy across varying complexities of the latent editors and region selectors, addressing limitations regarding manual intervention, training complexity, and over- and under-selection of regions along the way. The CORAL strategy can also enhance interactive editing experience by utilizing the predicted masks at each layer.

[3] as per Section 3.2 of their paper

References

[1] Yuval A., Or Patashnik, and Daniel Cohen-Or. Only a matter of style: age transformation using a style-based regression model. *ACM Trans. Graph.*, 40(4):45:1–45:12, 2021.

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

[3] Rameen Abdal, Peihao Zhu, John Femiani, Niloy J. Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions. In *SIGGRAPH*. ACM, 2022.

[4] Rameen Abdal, Peihao Zhu, Niloy J. M., and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3):21:1–21:21, 2021.

[5] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *ECCVw*, 2022.

[6] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022.

[7] Amit H. B., Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Or Patashnik, and Daniel Cohen-Or. State-of-the-art in the architecture, methods and applications of stylegan. *Comp. Graph. Forum*, 41(2):591–611, 2022.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of GANs. *CVPR*, 2020.

[10] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE TPAMI*, 2022.

[11] H. Dong, Simiao Yu, Chao Wu, and Y. Guo. Semantic image synthesis via adversarial learning. *ICCV*, pages 5707–5715, 2017.

[12] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Trans. Graph.*, 41(4):141:1–141:13, 2022.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi M., Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C., and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. In *NeurIPS*, 2020.

[15] Xianxu Hou, Linlin Shen, Or Patashnik, Daniel Cohen-Or, and Hui Huang. FEAT: face editing with attention. *CoRR*, abs/2202.02713, 2022.

[16] Omer Kafri, Or Patashnik, Yuval A., and Daniel Cohen-Or. Stylefusion: Disentangling spatial segments in stylegan-generated images. *ACM Trans. Graph.*, Mar 2022.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE TPAMI*, 43(12):4217–4228, 2021.

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

[20] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in GAN for real-time image editing. In *CVPR*, 2021.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[22] Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar Yanardag. Stylemc: Multi-channel based fast text-guided image generation and manipulation. In *WACV*, 2022.

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561. IEEE Computer Society, 2013.

[24] Seung H. L., Won Kyoung Roh, Wonmin Byeon, Sang Ho Yoon, Chan Y. K., Jinkyu Kim, and Sangpil Kim. Sound-guided semantic image manipulation. *CVPR*, 2022.

[25] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip H. S. Torr. Manigan: Text-guided image manipulation. In *CVPR*, 2020.

[26] Yahui Liu, Marco De Nadai, Deng Cai, Huayang Li, Xavier Alameda-Pineda, N. Sebe, and Bruno Lepri. Describe what to change: A text-guided unsupervised image-to-image translation approach. *ACMMM*, 2020.

[27] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, 2010.

[28] Seonghyeon Nam, Yunji Kim, and S. Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *NeurIPS*, 2018.

[29] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

[30] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit B. Patel, and Animashree Anandkumar. Semi-supervised stylegan for disentanglement learning. In *ICML*, 2020.

[31] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E. Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and CLIP. *CoRR*, abs/2107.12518, 2021.

[32] Gaurav Parmar, Yijun Li, Jingwan Lu, Richard Zhang, Jun-Yan Zhu, and Krishna Kumar Singh. Spatially-adaptive multilayer selection for GAN inversion and editing. In *CVPR*, 2022.

[33] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *CVPR*, 2022.

[34] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

[35] Justin N. M. Pinkney and Chuan Li. clip2latent: Text driven sampling of a pre-trained stylegan using denoising diffusion and CLIP. *BMVC*, 2022.

[36] Alec Radford, Jong W. K., Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[38] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[39] S. Reed, Zeynep Akata, Xinchen Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[40] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In *CVPR*, 2021.

[41] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[42] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021.

[43] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3d control over portrait images. *CVPR*, 2020.

[44] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021.

[45] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. *CVPR*, 2021.

[46] T. Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and X. He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. *CVPR*, 2018.

[47] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[48] Han Zhang, T. Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE TPAMI*, 41:1947–1962, 2019.

[49] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

[50] Yufan Zhou, Ruiyi Zhang, Jiuxiang Gu, Chris Tensmeyer, Tong Yu, Changyou Chen, Jinhui Xu, and Tong Sun. Tigan: Text-based interactive image generation and manipulation. In *AAAI*, 2022.