IT UNIVERSITY OF CPH

IMAGE EDITING WITH DENOISING DIFFUSION MODELS AND GRADIENT GUIDANCE

Johan Ødum Lundberg

Master thesis
Data Science
Johan Ødum Lundberg
Course Code: KISPECI1SE

September 1, 2023
Supervisor: Stella Grasshof
Supervisor: René Haas

September 1, 2023

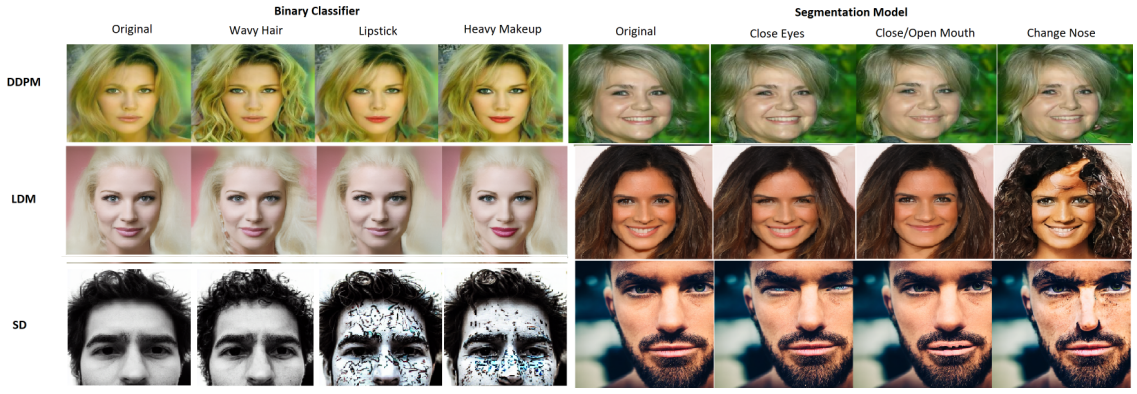

Figure 1: **Classifier and segmentation network-guided edits**. I present two guidance functions for editing images generated by DDMs. Here, I demonstrate some of the results of each guidance function using three different pretrained DDMs. Left: A binary attribute classifier uses its logit predictions to enable edits such *Wavy Hair*, *Lipstick* and *Heavy Makeup*. Right: A segmentation network uses its pixel predictions to edit facial features such as *eyes*, *mouth* and *nose*.

Abstract

The use of gradient guidance for image editing using pretrained Denoising Diffusion Models (DDMs) remains under-explored. In this work, I study the application of gradient guidance methods for editing images via pretrained DDMs. I present three gradient guidance methods: colour guidance, segmentation network guidance, and binary attribute classifier guidance. I start by demonstrating successful edits of synthetic images. Next, I explore challenges related to the general method, notably the selection of hyper-parameters, confirming findings from earlier studies. I also propose a regularization method for attribute entanglement and disentanglement, showing encouraging outcomes using the binary attribute classifier. Furthermore, I evaluate the ability of the classifier to edit an attribute correctly and find it to have the correct effect on the target attribute with some effect on correlated attributes. Lastly, I edit real images using the classifier and find the results to be less convincing. Overall, I have presented novel methods for image editing using pretrained DDMs, an area receiving little attention, and results showing some promise with further room for improvement.

Contents

1 Introduction

Denoising diffusion models (DDMs) [31] are fast becoming the primary method for editing images due to their ability to synthesize high-quality images. Notably, they have been shown to outperform Generative Adversarial Networks [8] in unconditional synthesis [5]. The great performance of DDMs motivates control of the generative process. This introduces the role of gradient guidance. Gradient guidance is a method for conditioning the generative process of the DDM on the gradient of a guidance function, thereby adding control. This method is particularly important for the control of pretrained and unconditional DDMs.

However, while DDMs continue to gain traction, the area of guidance remains under-explored. The under-exploration stems from both previous difficulties in applying guidance and the success of large text-driven DDMs. Specifically, models such as GLIDE [22], Imagen [29], DALL-E [25] and Stable Diffusion (SD) [26] yield highly diverse and state-of-the-art quality images. Further supporting this focus on text-based DDMs is their impressive editing and personalization capabilities as demonstrated in [10], [7], [28], [14], [34] and [3].

However, recent work by [2] proposed a universal guidance algorithm that addresses a previously encountered limitation of guidance. Specifically, their approach removes the necessity of retraining the DDM and guidance function. Nevertheless, there remains little published work on its use.

Prior work tends to study how guidance impacts the balance between image quality, image fidelity and adherence to the guidance method. Central to this balance is the choice of two hyperparameters; the guidance strength λ and conditioning range $[t_1, t_2]$ [4], [21], [15], [2]. These hyperparameters are essential to the effectiveness of the guidance function. Yet, how to select them remains unclear and they are currently determined based on qualitative results.

Given these challenges and gaps in the area of guidance of pretrained DDMs, I consider it worthwhile to press for further investigation. With this motivation, I propose the following.

In this paper, I propose three novel guidance functions to bridge the knowledge gap of how to edit images using pretrained DDMs. Specifically, I examine the use of colour guidance, segmentation network guidance and binary attribute classifier guidance. Each method is explored in the context of facial images for a rigorous analysis. The rest of this thesis is structured as follows.

First, in section 4.1, I explore the qualitative effect of each guidance function and demonstrate successful edits. Figure 1 illustrates the main results of two of the guidance functions[1]. Additionally, I provide a regularization method for entangling and disentangling attributes and demonstrate the results using the binary attribute classifier.

Second, in section 4.3, I carry out an analysis to shed light on how to optimally select the guidance strength λ and conditioning range $[t_1, t_2]$. I find the optimal choice to vary between the parameters and to depend on the facial attribute. Further, I find extending the conditioning range to have little effect.

Third, in section 4.4, I assess the feasibility of using the binary attribute classifier as a guidance function by measuring its effect on the target attribute and possibly correlated attributes. I find the classifier to have the intended effect, but that correlated attributes are affected.

Lastly, in section 4.4, I show the results of editing real images, finding the attribute classifier to be less effective.

To my knowledge, this study fills a gap in the research on universal guidance functions for synthetic image editing.

In summary, my contributions are as follows:

- I propose three universal guidance functions for editing images generated by pretrained models, one using colour, a segmentation network and a binary attribute classifier. I demonstrate, qualitatively, the feasibility of each and show several challenges related to their use. By doing so, I highlight possible new research directions and add to the current body of knowledge. Additionally, I demonstrate a regu-

[1] Colour guidance left out for visualization purposes.

larization method for disentangling and entangling different facial attributes using the binary attribute classifier.

- I find the choice of the conditioning range $[t_1, t_2]$ and guidance strength λ to vary interdependently and to depend on the target attribute, and that increasing the conditioning range has little effect.
- I demonstrate the effectiveness of the binary classifier measured by its attribute accuracy and average logit change calculated from the edit of 100 images. Notably, the results indicate that the edit of an attribute affects the target attribute, but it also affects correlated attributes.
- Lastly, I find the binary classifier and segmentation model to have less of an effect when editing real images.

2 Related Work

In this section, I review common methods of guiding a DDM.

2.1 Guided Diffusion

One line of research in image editing with unconditional models has focused on using guides. For instance, [4] conditions the synthesis of an image on a reference, such as another image or pixel guide. They go about this by ensuring that the down-sampled representation of the image stays close to the down-sampled representation of the reference. Their work showed that generated samples deviate from the guide if a conditioning range of 500 steps or less is used.

Similarly, [21] adds coarse coloured pixel strokes to the image and solves a reverse stochastic differential equation. They obtained a reasonable conditioning range using a combination of both user feedback and binary search. Importantly, they found that a shared conditioning range works well for similarly guided tasks.

Together, these studies explore global image editing, without control of details.

2.2 Classifier-Guidance

Classifier guidance, also known as gradient guidance, emerged as a promising method in [5]. This work illustrated that an unconditional DDM could be conditioned on a classifier. Specifically, the prediction of the DDM would be conditioned on the label prediction of the classifier. However, classifier guidance tends to require retraining of the classifier for optimal performance. This is primarily due to the classifier being trained on clean images while the DDM generates noisy images. Most studies, like those in [19], [1], [22], and [15], have addressed this by finetuning either the DDM or the classifier. For instance, [22] and [15] both guide a diffusion model using Contrastive Language-Image Pretraining (CLIP) [24], but not before finetuning the CLIP model on noisy images. Collectively, this research employs language, image, and multimodal guidance. In comparison to my work, these studies focus on global edits, with less control of the details.

A recurring theme is the search for a balance between image fidelity and faithfulness to the classifier. Taken together, this is done by finding the optimal conditioning range and guidance strength. Currently, the consensus appears to be that too narrow a range results in an image non-faithful to the classifier while too wide a range means the classifier takes over and affects the realism of the final image. Findings also suggest that the conditioning range should be adjusted depending on the attributes.

2.3 Universal Guidance

A recent study [2], has proposed a variation of classifier-guidance that accepts any form of conditioning, such as a segmentation network or classifier. They call it universal guidance. Their method leverages a clean

Figure 2: **Forward & reverse process overview.** A DDM consists of a forward and reverse process. The forward process adds Gaussian noise to the input image $\mathbf{x}_0$ in T time steps resulting in white Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse process samples from a Gaussian distribution and generates an image in T time steps.

version of a noisy image to convey more useful information to the diffusion model, thereby increasing control and avoiding any retraining.

They demonstrated their method by successfully generating images guided by both a segmentation network and classifier, motivating the use of their approach in my work.

2.4 Classifier-Free Guidance

The paper [12] introduces the concept of classifier-free guidance (CFG). This involves training a DDM with and without labels some percentage at the time. Consequently, the unconditional and conditional noise predictions can be linearly interpolated during sampling using a hyperparameter; *guidance strength*. This leaves out the requirement of having to train a separate classifier, however, it also binds the DDM to a particular kind of guidance. CFG has proven quite a convenient method of training and is often used in text-driven DDMs to balance quality with prompt fidelity.

2.5 H-Space

Recent evidence suggests that DDMs have their own semantic latent space called h-space [16], specifically in the bottleneck of a U-Net [27]. One recent study [9], demonstrated the use of a binary attribute classifier for discovering semantically interpretable directions within h-space. They showed that image samples annotated using the classifier can be used to find corresponding editing directions using simple vector arithmetic. They did this by sorting the annotated samples by their attribute scores, in descending order. After sorting, they found an editing direction q by taking the difference between the latent representations of the samples with the highest scores and those with the lowest scores.

3 Methods

In this section, I provide an overview of diffusion models and then describe my guidance function method followed by a detailed description of each guidance function.

3.1 Denoising Diffusion Models

The denoising diffusion model is a latent variable deep generative model that learns to model the data distribution $p(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$. It consists of two processes; a fixed forward process that gradually adds noise to the input image $\mathbf{x}_0$ in T time steps and a parameterized reverse process that iteratively removes the noise, see figure 2. The noise added during the forward process is centred around a Gaussian with parameters controlled by a noise schedule $\beta_t \in [0, 1]$ with $\beta_1 < \beta_2 < \cdots < \beta_{T-1} < \beta_T$ such that $\mathbf{x}_T \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$.

The schedule can be further defined as $\alpha_t = \prod_{s=1}^{t} \alpha_s$ with $\alpha_s = 1 - \beta_t$. A noisy image is then generated by the forward process as

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\mathbf{n}, \tag{1}$$

with $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Note, equation 1 can represent a single step of the forward process or multiple steps of the forward process.

Beginning by sampling a Gaussian $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$, the reverse process slowly generates a clean image $\mathbf{x}_0$ by removing the added noise resulting in a target distribution

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t \geq 1} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{2}$$

Here, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ represents the prediction of a slightly less noisy image, a single step of the revere process in figure 2. Using the definition of the reverse process as defined in [32], a single step is defined as

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{P}_t(\epsilon_t^\theta(\mathbf{x}_t)) + \mathbf{D}_t(\epsilon_t^\theta(\mathbf{x}_t)) + \sigma_t \mathbf{z}_t. \tag{3}$$

The term ϵ_t^θ represents the noise prediction at time step t of a neural network parameterized by θ. The neural network is U-Net trained to predict the noise $\mathbf{n}$ that was added to $\mathbf{x}_t$. For brevity, I denote $\mathbf{P}_t(\epsilon_t^\theta(\mathbf{x}_t))$ as $\mathbf{P}_t$ and $\mathbf{D}_t(\epsilon_t^\theta(\mathbf{x}_t))$ as $\mathbf{D}_t$. $\mathbf{P}_t$ is defined as the predicted $\mathbf{x}_0$ at time step t and $\mathbf{D}_t$ is defined as the direction pointing to $\mathbf{x}_t$ at time step t. They can be defined as

$$\mathbf{P}_t = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_t^\theta(\mathbf{x}_t)}{\sqrt{\alpha_t}}, \tag{4}$$

$$\mathbf{D}_t = \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_t^\theta(\mathbf{x}_t). \tag{5}$$

The term σ_t controls the amount of noise added at each step of the reverse process. It is in turn controlled by the hyperparameter η and defined as

$$\sigma_t = \eta_t \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}. \tag{6}$$

When $\eta = 0$, the standard deviation $\sigma_t = 0$ and the reverse process corresponds to the deterministic Denoising Diffusion Implicit Model (DDIM) [32] mapping $\mathbf{x}_t$ to a unique $\mathbf{x}_0$. Conversely, $\eta = 1$ corresponds to the stochastic DDPM [11].

As editing requires access to a latent code $\mathbf{x}_t$, an inversion method for real-image editing was suggested for the DDIM ([32] [5]). Given $\sigma_t = 0$, in the limit of small steps, $\mathbf{x}_T$ can be recovered from $\mathbf{x}_0$ using

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}}\mathbf{P}_t + \sqrt{1-\alpha_{t+1}}\epsilon_t^\theta(\mathbf{x}_t). \tag{7}$$

However, this is an approximation of $\mathbf{x}_T$ and usually requires around 1000 time steps to be accurate. Another inversion method was recently proposed by [13]. They defined an edit-friendly latent noise space that makes it possible to extract both $\mathbf{x}_t$ and each $\mathbf{z}_t$. Specifically, they define the forward process as

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\tilde{\epsilon}_t, \quad 1, \ldots, T, \tag{8}$$

where the $\tilde{\epsilon}_t$ are statistically independent. Note, in the paper they use $\bar{\alpha}_t$ where I use α_t. While this forward process is slower compared to equation 1, it allows for perfect reconstruction of the image $\mathbf{x}_0$ and starting from any time step t. Given this forward process, they extract each $\mathbf{z}_t$ as

$$\mathbf{z}_t = \frac{\mathbf{x}_{t-1} - \hat{\mu}_t(\mathbf{x}_t)}{\sigma_t}, \tag{9}$$

where each $\hat{\mu}_t(\mathbf{x}_t)$ is equal to equation 3 with $\eta = 0$. Using this method, an image $\mathbf{x}_t$ can be edited by modifying the $\mathbf{z}_t$s of equation 3.

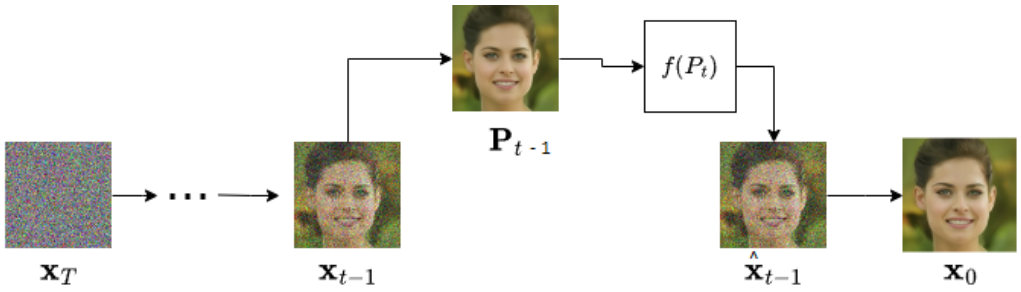

Figure 3: **High-level method overview**. During the synthesis of an image, $\mathbf{x}_{t-1}$ is edited by adding the gradient of a loss L with respect to $\mathbf{x}_{t-1}$, scaled by α_t^2, to $\mathbf{x}_{t-1}$. The loss is found using a guidance function f taking $\mathbf{P}_{t-1}$ as input.

3.2 Established Gradient Guidance Methods

Gradient guidance, as introduced in [5], conditions the unconditional reverse process of the DDM on a classifier trained to predict a label y. Specifically, given a step of the reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and a classifier $p_\phi(y|\mathbf{x}_t)$, each step is refined as

$$p_{\theta,\phi}(\mathbf{x}_{t-1}|\mathbf{x}_t, y) = Z p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) p_\phi(y|\mathbf{x}_t). \tag{10}$$

Here, ϕ represents the parameters of the classifier and Z is a normalization constant. In the case of a denoising diffusion implicit model (DDIM), the above can be approximated as

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\Big(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\hat{\epsilon}}{\sqrt{\alpha_t}}\Big) + \sqrt{1-\alpha_{t-1}}\hat{\epsilon}. \tag{11}$$

This is equation 3 with $\eta = 0$. The noise $\hat{\epsilon}$ is a modification of the original noise prediction of the DDM and is taken to be

$$\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t) - \sqrt{1-\alpha_t}\nabla_{\mathbf{x}_t} log\, p_\phi(y|\mathbf{x}_t). \tag{12}$$

More specifically, the noise prediction of the DDM $\epsilon_\theta(\mathbf{x}_t)$ is modified using the gradient of the prediction of the classifier, which is used in the sampling step of the DDIM, equation 11.

The work by [2], introduces a variation of the above approach, called universal guidance. They introduce the term of a guidance function f to mean any differentiable function, thus no longer restricting guidance to the use of a classifier. The guidance function takes as input $\mathbf{P}_t$, rather than $\mathbf{x}_t$, to provide meaningful guidance to a loss function l. With this, equation 12 is rewritten as

$$\hat{\epsilon}_\theta(\mathbf{x}_t) = \epsilon_\theta(\mathbf{x}_t) + s(t) \cdot \nabla_{\mathbf{x}_t} l(c, f(\mathbf{P}_t)). \tag{13}$$

Here, c signifies a class label and $s(t)$ defines a time step-dependent scalar.
With this, I introduce my method of gradient guidance, which is similar to equation 13.

3.3 Proposed Guidance Framework

Given a clean image $\mathbf{x}_0$ and its corresponding noisy latent $\mathbf{x}_t$, the goal is to change a property of the image by modifying the latent at each step of the reverse process. The purpose of the guidance function is to quantify a descriptive property of the image, which, in this study is referred to as an attribute score. The attribute can constitute visual characteristics, semantic features, style etc. Using the attribute score, we can iteratively nudge $\mathbf{x}_t$ in the direction of an edit $\mathbf{x}'_0$.

Algorithm 1 Iterative Guidance Function Editing

Input: Image $\mathbf{x}_t$, guidance strength λ, time step t_1, time step t_2, guidance function f, binary mask $\mathbf{M}$
Output: Image $\mathbf{x}_0$

for $t = T, \dots, 1$ **do**
  if $t < t_1$ **or** $t >= t_2$:
    return $\mathbf{x}_t$
  $\mathbf{x}_{t-1} = p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
  $s = f(\mathbf{P}_{t-1})$
  $L = s \cdot \lambda$
  $\nabla L = -\nabla_{\mathbf{x}_{t-1}}(L)$
  Optionally: $\nabla L = \mathbf{M}\nabla L$
  $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} + \nabla L \cdot \alpha_t^2$
end for
return $\mathbf{x}_0$

Let f be a guidance function and $s = f(\mathbf{P}_t)$ be an attribute score s. The loss is then taken to be $L = s \cdot \lambda$ with λ being guidance strength. A nudge of $\mathbf{x}_t$ is then done as $\mathbf{x}_t = \mathbf{x}_t + \nabla_{\mathbf{x}_t}(L) \cdot \alpha_t^2$. See figure 3 for a high-level overview.

To ensure a successful edit and the fidelity of $\mathbf{x}_0'$ to $\mathbf{x}_0$, a conditioning range $[t_1, t_2]$ and guidance strength λ must be found. The conditioning range defines a sequence of time steps at which to apply the guidance function with t_1 being the first of the time steps and t_2 being the last. Throughout this thesis, the terms t_1 and t_2 are defined as indices of the reverse process. Meaning, the index $t_1 = 0$ indicates time step T, $t_1 = 1$ implies time step $T - 1$ and $t_1 = T - 1$ means time step 1. However, the convention is to refer to t_1 and t_2 as the actual time steps.

For additional control, a binary mask $\mathbf{M}$ can be used to replace values in $\nabla_{\mathbf{x}_t}(L)$ by $\mathbf{M}\nabla_{\mathbf{x}_t}(L)$. This will ensure that changes to $\mathbf{x}_t$ only occur within the area of the image as specified by the mask. I define the algorithm for editing $\mathbf{x}_t$ with a guidance function in algorithm 1.

3.3.1 Colour Guidance

To begin, I define a guidance function f to edit the colours of $\mathbf{x}_0$. Given $\mathbf{P}_t$, a colour channel c, and a target value w, the attribute score s is computed as $s = f(\mathbf{P}_t, c, w) = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{P}_{t,i,c} - w|$ with i representing a pixel. The attribute score s represents the mean absolute difference between the colour channel c of the image $\mathbf{P}_t$ and the target value w. This task should be straightforward for the DDM since any modifications incurred by the guidance function are expected to be superficial, making no changes to the core semantics of the image. For instance, changing the age should make the resulting image $\mathbf{x}_0' = f_{age}(\mathbf{P}_t)$ more distinct from $\mathbf{x}_0$ compared to $\mathbf{x}_0' = f_{colour}(\mathbf{P}_t)$.

3.3.2 Segmentation Network Guidance

For a more nuanced approach, I utilize a segmentation network[2] to make semantic changes such as opening and closing the eyes or mouth. It is a network based on [37] and pretrained on CelebAMask-HQ [17]. Specifically, it is trained on 19 different facial attributes: *"Background", "Skin", "Left brow", "Right brow", "Left eye", "Right eye", "Eyeglasses", "Left ear", "Right ear", "Earring", "Nose", "Mouth", "Upper lip", "Lower lip", "Neck", "Necklace", "Cloth", "Air", "Hat"* .

Let $E_S(\cdot)$ be the segmentation network and let $\mathbf{p}$ be the pixel predictions of the segmentation model given by

$$\mathbf{p} = E_S(\mathbf{P}_t). \tag{14}$$

[2]https://github.com/zllrunning/face-parsing.PyTorch

Here, $\mathbf{p}$ is of shape $C \times H \times W$ where C defines the number of classes that the network is trained on. To obtain an attribute score s_i for a class of interest $i \in \{1, 2, \ldots, C\}$, I apply the softmax on $\mathbf{p}$ along the C dimension and sum across the H and W dimension as:

$$s_i = \frac{1}{256 \times 256} \sum_{j=0}^{255} \sum_{k=0}^{255} \frac{\exp(\mathbf{p}_{c,h,w})}{\sum_j \exp(\mathbf{p}_{c,h,w})}. \tag{15}$$

In the case of k classes of interest, the scores are summed $\sum_i^k s_i$. For instance, to open or close both eyes, the score for each eye would be combined.

3.3.3 Binary Attribute Classifier Guidance

While the segmentation network provides a method for editing facial features, it is limited to the labels of the network. As the labels of the network correspond to image segments, the guidance function cannot change semantic attributes such as *age*, *gender* and *attractive*.

For this reason, I explore the use of a binary attribute classifier. I use the attribute classifier from the paper *AnycostGAN* [18]. It is a classifier based on a resnet50 backbone trained to predict 40 attributes on CelebA [20]. I take the attribute score s to be the negative or positive prediction of an attribute logit prediction p_i. However, in some experiments, I took the score to be the maximum of the attribute prediction, see appendix F for these results.

3.4 Masking

The creation of the mask $\mathbf{M}$ is done as follows. Given segmentation network $E_S(\cdot)$ as described in section 3.3.1, the segmentation map of the image $\mathbf{x}_0$ is found as $\mathbf{S} = E_S(\mathbf{x}_0)$ with $\mathbf{S} \in \{1, \ldots, 18\}^{H \times W}$. Let $y_i \in \{1, \ldots, 18\}$ represent a class of interest from the set of classes predicted by the segmentation network, the mask $\mathbf{M}$ is then defined as:

$$\mathbf{M}[j, k] = \begin{cases} 1 & \text{if } \mathbf{S}[j, k] = y_i \\ 0 & \text{else,} \end{cases} \tag{16}$$

with $\mathbf{S}[j, k]$ defining a single pixel prediction.

4 Experiments

4.1 Experiment Setup

Diffusion models. In my experiments, I use three pretrained DDMs from the Diffusers library [35]. The first[3] (DDPM), is an unconditional model trained on the CelebA data set at resolution 256×256. The second[4] (LDM), is an unconditional latent model [26], also trained on CelebA at resolution 256×256. However, the diffusion process occurs in the latent space of an encoder. The last[5] (SD), is a conditional text-to-image latent model, Stable Diffusion version 1.4, trained on a subset of the LAION-5B data set [30] at resolution 512×512. For each model, I use a DDIM scheduler with parameters instantiated from the pretrained model's configuration file.

Implementation details. Unless specified otherwise, I use $\eta = 1$ and 50 inference steps. In the case of the DDPM, I set clip_sample $= True$ for synthetic image editing and clip_sample $= False$ for real image editing as editing would not work properly otherwise. When *True*, the prediction $\mathbf{P}_t$ is clipped between 0 and

[3] https://huggingface.co/fusing/ddpm-celeba-hq
[4] https://huggingface.co/CompVis/ldm-celebahq-256
[5] https://huggingface.co/CompVis/stable-diffusion

1. When editing images, the guidance strength λ and conditioning range $[t_1, t_2]$ are hyperparameters that I manually adjust for each experiment.

Experiments using SD rely on a text-prompt and a classifier-free guidance hyperparameter, *CFG scale*. The *CFG scale* controls the balance between the image quality and prompt fidelity. Most experiments use a *CFG scale* of 3.5-7.0, as this tends to give good results.

Evaluation Metrics. In quantitative experiments, I evaluate the results using three evaluation metrics: the Learned Perceptual Image Patch Similarity (LPIPS) [38], classifier accuracy, and the average change in the negative and positive logit directions for the classifier.

LPIPS: This metric measures the perceptual similarity between the edited and original images. A lower LPIPS score indicates that the edit aligns more closely with the original.

Classifier Accuracy: This metric measures the percentage of edits that did not result in a change to an attribute. Low accuracy of a target attribute with a high accuracy for the remaining attributes indicates a successful edit that did not affect other attributes except for the target attribute. Contrarily, a low accuracy for every attribute or a high accuracy for the target attribute can be considered failure cases.

Average Logit Change: While classifier accuracy measures the success of an edit, it does not necessarily specify which attribute was affected the most. I address this by calculating the average logit change in both the negative and positive direction for each attribute prediction.

A positive change in the negative direction of an attribute means this outcome is more likely while a negative change in the positive direction means this outcome is less likely.

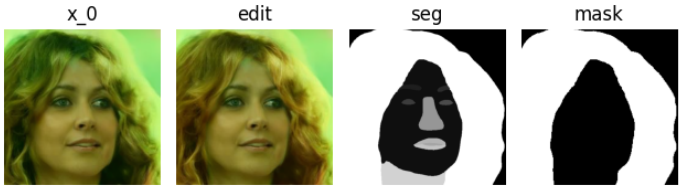

Figure 4: **Colour Guidance Example.** The figure shows the image $\mathbf{x}_0$, edited image $\mathbf{x}'_\mathbf{0}$ and the segmentation map and mask created from the segmentation network. In this example, the image was generated using the DDPM with 100 inference steps and was edited with colour guidance. A guidance strength of 500 and a target value of 0.9 was used.

4.2 Qualitative Experiments

In this section, I demonstrate the qualitative results of editing synthetic images. Images denoted by *original* represent the initial image prior to editing.

4.2.1 Colour Edits

I demonstrate the effect of colour guidance. Figure 4 shows an example of editing the colour of the hair by masking out the gradients ∇L. In this case, the guidance function was used at every step of the reverse process consisting of 100 inference steps with a guidance strength $\lambda = 500$. The choice of guidance strength and conditioning range may depend on multiple factors such as the image itself, the number of inference

steps and the DDM. For instance, in figure 5, each DDM generates an image where the colour of the hair has been changed. It shows how the choice of the hyperparameters can work for one model, but not for another.

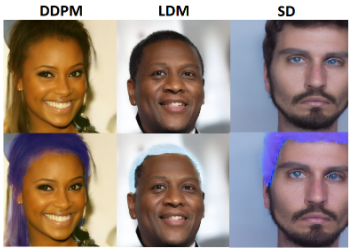

Figure 5: **Impact of the choice of DDM and hyperparameters.** The figure shows the impact of the choice of DDM and the hyperparameters; guidance strength λ and conditioning range $[t_1, t_2]$. Each image was edited using colour guidance with a guidance strength of 5000 and a guidance range of $[0, 30]$. SD used the text prompt: "A close up id photo of a man" and a classifier-free guidance of 3.5. Each DDM used 50 inference steps.

Additionally, figure 6 illustrates the variable guidance strength values needed for editing two DDPM-generated images with colour guidance. Since an increase in the number of inference steps typically requires a decrease in the guidance strength, this example not only emphasizes the difference but also hints at potential biases in the model, likely stemming from the training data.

To demonstrate a working example for the SD model, figure 7 shows the result of applying the attribute guidance at every step of a 50-step inference process with a guidance strength $\lambda = 500$.

See appendix G for additional colour edits.

4.2.2 Segmentation Network Guidance

I present the results of segmentation network guidance. As shown in figures 8 and 9, the labels of the segmentation network correspond to edits of the corresponding image segments. The edits include actions such as opening or closing the mouth or eyes, resizing the nose, and changing the thickness of the hair. In the case of changes made to the eyes, the score for each eye was combined into a single score. Similar edits can also be made to images generated by the SD model. For instance, figure 10 shows an example of opening and closing the eyes without a mask.

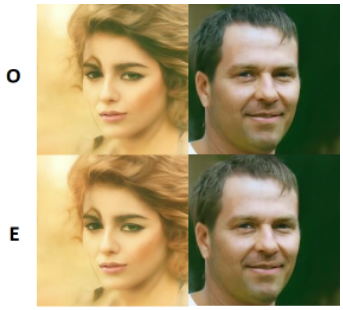

Figure 6: **Illustration of variable guidance strengths and results.** The figure shows two images generated by the DDPM and edited using colour guidance. Left: 50 inference steps and a guidance strength of 1000. Right: 100 inference steps and a guidance strength of 50000. The guidance function was applied at every step of the synthesis process.

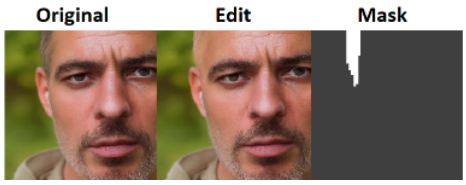

Figure 7: **SD colour guidance example.** Image sampled from SD with 50 timesteps and a classifier-free guidance scale of 3.5. The prompt was "Photo of a man from the front looking at the camera, close up, id photo". Using the colour attribute function at every step, the colour of the hair is changed to be more red using a guidance strength of 500 and a target value of $w = 0.9$.

While a mask $\mathbf{M}$ was used for most of the edits in figure 8, this is not the case in figure 9, showing the feasibility of the method without a mask. On the other hand, the edits made in figure 8 are minor and show the difficulty of editing these particular attributes using this method.

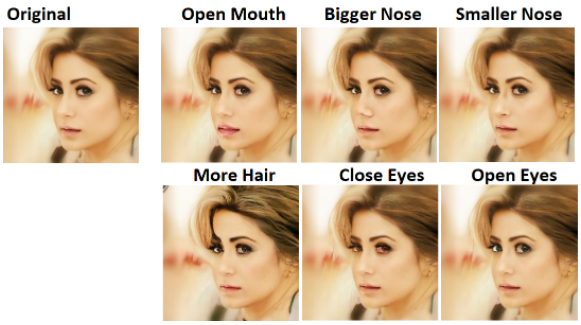

Figure 8: **Segmentation network guidance with a DDPM.** The images were generated using 100 inference steps and $\eta = 1$. The figure shows, starting from the upper left: The original image, open mouth, bigger nose, smaller nose. Starting from the bottom left: Hair change, close eyes and open eyes. Except for the hair example, ∇L was masked out with the corresponding mask.

Further, figure 9 indicates that different facial attributes require different hyperparameter values to make a successful edit. The figure also shows that steps are assigned after the last conditioning step t_2. This was to improve the quality of the image after the edit. Additionally, a conditioning range $[26, 75]$ was chosen based on an observation that the model tends to synthesize details, such as the mouth, around $t_1 = 26$.

In general, I have found the LDM to be more susceptible to the segmentation network guidance and the DDPM less so. Taken together, the results of the segmentation network indicate room for further improvement and that images do not necessarily respond to the guidance.

4.2.3 Binary Attribute Classifier Guidance

Similarly to the segmentation network, the binary classifier can be used to make edits corresponding to its labels. This is illustrated in figures 11 and 12 where edits correspond to the labels *Eyeglasses*, *Age* and *Big Nose*. From my experiments, I have found edits to affect other correlated attributes. For instance, adding eyeglasses can increase the age. See Appendix E for further details. Also, in figure 12 we see that adding eyeglasses increases the size of the nose.

Such entanglement of attributes is common in generative models. In [9] they solved this by removing from a vector $\mathbf{v}_1$, representing the editing direction, the projection of it onto an unwanted direction $\mathbf{v}_2$. To disentangle age and eyeglasses, I define a new score that incorporates a regularization term for eyeglasses

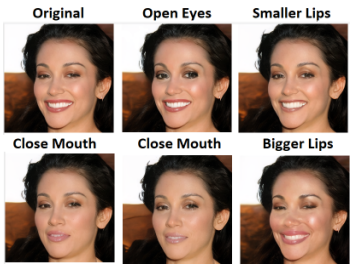

Figure 9: **Segmentation network guidance with an LDM.** The images are, starting from the top left, Original, Open Eyes (guidance strength: -5000, $t_1 = 70, t_2 = 79$), Smaller lips using the lips minus the mouth as a loss (guidance strength: 5000, $t_1 = 26, t_2 = 75$), Close mouth using the mouth as the loss (guidance strength: 5000, $t_1 = 26, t_2 = 75$), Close mouth using the lips as the loss (guidance strength: 5000, $t_1 = 26, t_2 = 75$), Bigger lips using the mouth minus the lips as a loss (guidance strength: 2000, $t_1 = 26, t_2 = 75$).

Figure 10: **Segmentation network guidance using SD with varying guidance strengths.** The figure shows edits of the eyes with the original image in the middle, closed eyes to the left and opened eyes to the right. The images were generated using 100 inference steps, the prompt "*A man looking at the camera, close-up, id-photo*" and a CFG scale of 7.0. The conditioning range was $[70, 79]$. The guidance strengths were, starting from the left, 1500, 1000, -500 and -1000.

into the score for age. Specifically, I define it as

$$s = s_{age,\mathbf{P}_t} + (s_{eyeglasses,\mathbf{P}_t} - s_{eyeglasses,\mathbf{x}_0})^2. \tag{17}$$

Here, $s_{eyeglasses,\mathbf{P}_t}$ signifies the eyeglasses score of $\mathbf{P}_t$ at time step t and $s_{eyeglasses,\mathbf{x}_0}$ is the eyeglasses score of the original image $\mathbf{x}_0$. Similarly, glasses can be added using

$$s = s_{age,\mathbf{P}_t} + (s_{eyeglasses,\mathbf{P}_t} + s_{eyeglasses,\mathbf{x}_0})^2. \tag{18}$$

The result of this is shown in figure 13.

Taken together, the qualitative results show both successful edits and less successful edits with definite challenges of applying each guidance function. These challenges include finding the proper hyperparameters for each image and model, and there may not always be a proper setting. Lastly, it should be noted that while the DDPM and LDM have been trained on face images, SD has been trained on my types of images, which might make it more difficult for the model to apply guidance.

4.3 Analysis of the Guidance Strength and Conditioning Range

As has been demonstrated, the optimal choice of guidance strength λ and conditioning range $[t_1, t_2]$ can vary between images, models and guidance functions, but also interdependently. Arbitrarily increasing λ is not guaranteed to work properly as the model might struggle to remove the noise incurred from the nudge of $\mathbf{x}_t$. Likewise, expanding the guidance range will also not necessarily improve results. As shown in figure 14, guiding the DDM using the binary classifier with a conditioning range of $[0, 95]$ and a guidance strength of 5 did not have an effect until the final stage of the reverse process. It can be seen that in the early steps of

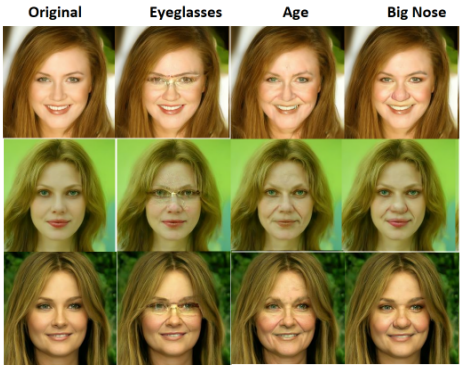

Figure 11: **Binary classifier guidance with a DDPM.** Using the binary classifier, we can edit attributes such as age and gender. Images are generated using 100 inference steps, a guidance range of $[0, 95]$ and a guidance strength $\lambda = 5$. The figure shows the original and edits of glasses, age and nose, respectively. See figure 22 in Appendix E for additional examples.

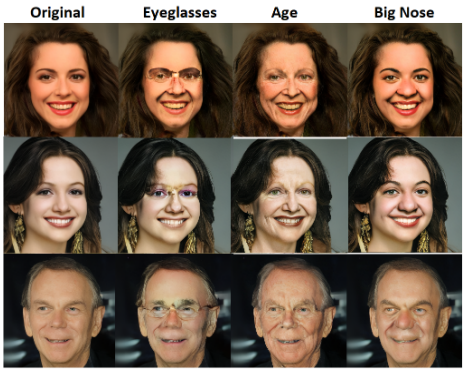

Figure 12: **Binary classifier guidance with an LDM.** Using the binary classifier, we can edit attributes such as age and gender. Images are generated using 100 inference steps, a guidance range of $[0, 95]$ and a guidance strength $\lambda = 5$. The figure shows the original and edits of glasses, age and nose, respectively. See figure 23 in appendix E for additional examples.

the reverse process, $\mathbf{P}_t$ is considerably more noisy. As such, the classifier and segmentation model might struggle to provide a useful signal to the DDM since they were trained on clean images. A problem similar to that in classifier-guidance. Additionally, we see that larger semantic changes begin to occur after time step 30 (4th image) and that details are added during the final stages, which is also where the edit takes effect.

Generally, the figure motivates the use of a shorter conditioning range, such as $[90, 95]$, and shows that a guidance strength $\lambda = 5$ is sufficient in this case.

To delve deeper, I carry out a grid search over possible parameter combinations. For this experiment, I use the DDPM and binary classifier. Limiting the scope of the experiment to this model and guidance function was based on computational limitations. To evaluate each parameter setting, I generate ten images and edit each image using the age attribute. I then compute the LPIPS metric for every edited image and take the average. For visualization purposes, I only show the result for one of the images.

As shown in figure 15, different conditioning ranges and guidance strengths can be used to make a successful edit. Successful edits have an average LPIPS around 0.4 indicating closer similarity to the original. In contrast, images with an LPIPS value above 0.4 start to get noisy pointing to their dissimilarity with the

Figure 13: **Disentanglement and Entanglement of age and eyeglasses.** The images are generated using the DDPM with 100 inference steps and edited with binary classifier guidance. The conditioning range was $[20, 75]$ and the guidance strength $\lambda = 600$.

Figure 14: **Average sample of $\mathbf{P}_t$ at different stages.** The figure shows the change in $\mathbf{P}_t$ during the synthesis of the edited image (on the right) using 100 inference steps. Each image of $\mathbf{P}_t$ represents an average of 10 samples of $\mathbf{P}_t$ starting from time steps 0-10 (left) and increasing with a step size of 10. In this example, the binary classifier was used to increase the person's age using a guidance range of $[0, 95]$ and a guidance strength of 5.

original. In some cases, the edit appears to have had an effect, however, it made the image noisy. This suggests the need for additional steps to remove the noise. In [16], they allocate some steps between a step t and the last step to reduce quality deficiency. I have also found this to be beneficial. In Appendix C, I provide additional details on this and in Appendix D, I show extra grid-search experiments.

All in all, this analysis arrives at previously known conclusions. Specifically, edits using guidance functions depend upon the guidance function itself, the targeted attribute, the DDM and the hyperparameters. Thus, this analysis serves to verify these conclusions in this particular setting.

4.4 Attribute Consistency

As has been shown, the application of the binary classifier could lead to the editing of correlated attributes. To ascertain how widespread this is, and to measure the editing feasibility of the method, I use the accuracy of the binary classifier as a measure of attribute consistency. In other words, if the accuracy of an attribute is close to 1, the attribute consistency is high. Furthermore, we would want to see poor accuracy for the attribute that is meant to be edited. Additionally, we would want to see the largest change in the targeted attribute and not in another correlated attribute. As such, I also measure the average change in the negative and positive logit prediction of each attribute.

For this experiment, I generate 100 images using the DDPM and edit each with the classifier's positive prediction for the attribute *youth* as the loss, which is meant to increase a person's age. I base my choice of the guidance strength λ and $[t_1, t_2]$ on the results of previous experiments. Specifically, I use $\lambda = 5$ and a conditioning range $[0, 95]$, although a shorter range could be used, as indicated in the prior experiment. Example edits are shown in figure 16.

As shown in table 1, the accuracy for the attribute *youth* is very low indicating that the attribute function often edits the image successfully. Additionally, the classifier is not performing well on the other attributes,

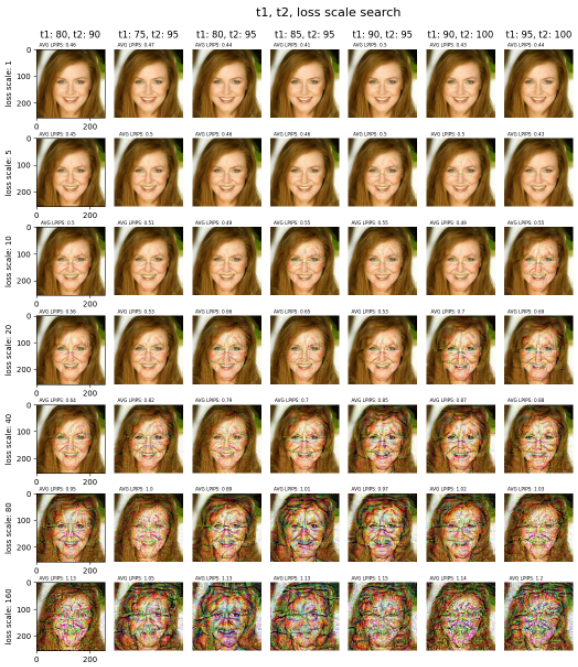

Figure 15: **Parameter Grid-Search.** The figure shows the result of editing an image with different λ and $[t_1, t_2]$ values. Next to each image is shown the average LPIPS calculated from the edit of 10 images. Here, loss scale refers to the guidance strength λ.

indicating they are correlated. This is, however, not necessarily problematic as *youth* is a concept that usually necessitates a change to attributes such as *Gray Hair*. I present additional results for other attributes in Appendix A, which show good results for most non-correlated attributes, I.e. a high accuracy.

Table 1: **Attribute consistency measured using the accuracy of the binary classifier.** The accuracy is calculated from the editing of 100 images. The images were generated using a DDPM with 100 inference steps. They were then edited using the positive prediction for the attribute *youth* of the binary attribute classifier. A guidance strength $\lambda = 5$ and conditioning range $[0, 95]$ were used. $\downarrow$ indicates that lower is better, for the target attribute.

	Attractive	Bags Under Eyes	Gray Hair	Heavy Makeup	Young
$\downarrow$ Acc.	0.21	0.43	0.71	0.58	0.08

From table 2, it can be seen that *youth* is the attribute that is on average affected the most. This can be seen by the increase in the *Negative* direction and decrease in the *Positive* direction of the *youth* attribute, meaning *youth* being false is more likely after the edit. Other attributes such as *Eyeglasses*, *Double Chin*, *Gray Hair* and *Bags Under Eyes* also tend to change, indicating that they are correlated with not being young, i.e. being old. Similarly, attributes such as *Attractive* and *Heavy Makeup* are determined as less likely. In the middle of the table, we find attributes with no correlation with either young or old, such as *5 o clock shadow* and *wearing earrings*.

Table 2: **Average logit change in the negative and positive direction per attribute sorted in descending order by the largest change to the *Negative* predictions.** The average change of each attribute is calculated from the average change of 100 images. This is done by calculating the difference between an attribute's original logit score and its edited logit score. This is then summed across images and divided by 100. The images were generated using a DDPM with 100 inference steps.

ID	Attribute	Negative	Positive
39	Young	5.06	-5.15
2	Attractive	3.13	-3.36
18	Heavy_Makeup	2.46	-2.57
36	Wearing_Lipstick	1.64	-1.98
25	Oval_Face	1.49	-1.12
9	Blond_Hair	1.33	-1.29
24	No_Beard	1.09	-1.06
11	Brown_Hair	0.65	-0.64
8	Black_Hair	0.64	-0.90
6	Big_Lips	0.61	-0.91
12	Bushy_Eyebrows	0.57	-0.55
32	Straight_Hair	0.55	-0.70
1	Arched_Eyebrows	0.50	-0.30
21	Mouth_Slightly_Open	0.46	-0.35
31	Smiling	0.32	-0.23
26	Pale_Skin	0.32	-0.49
23	Narrow_Eyes	0.18	-0.23
29	Rosy_Cheeks	0.16	0.04
27	Pointy_Nose	0.11	-0.29
0	5_o_Clock_Shadow	0.03	-0.05
34	Wearing_Earrings	-0.04	0.01
35	Wearing_Hat	-0.05	0.08
5	Bangs	-0.10	0.12
33	Wavy_Hair	-0.17	0.05
30	Sideburns	-0.27	0.30
37	Wearing_Necklace	-0.28	0.49
16	Goatee	-0.37	0.30
19	High_Cheekbones	-0.41	0.53
10	Blurry	-0.73	0.63
28	Receding_Hairline	-0.82	0.89
4	Bald	-1.09	1.02
38	Wearing_Necktie	-1.55	1.65
7	Big_Nose	-1.90	2.19
22	Mustache	-1.99	1.80
20	Male	-1.99	1.93
3	Bags_Under_Eyes	-2.04	1.89
13	Chubby	-2.05	2.16
15	Eyeglasses	-2.39	2.33
14	Double_Chin	-2.70	2.80
17	Gray_Hair	-3.34	3.26

Figure 16: Example images from the quantitative Experiment changing the age of each person using a DDPM with 100 inference steps and the binary classifier.

Taken together, the experiments on attribute consistency show that the use of the binary classifier to edit the age of a person works correctly without affecting non-correlated attributes. Furthermore, the target attribute *age* is the attribute affected the most. However, this is for a particular setting of the hyperparameters, which may not work as intended for a different attribute change.

4.5 Real Image Editing

Here, I show the results of editing real images using the DDPM and the binary attribute classifier. I do so using both the inversion method in equation 7 (DDIM) and the one in equation 9 (DDPM). In the case of the latter, this is used to extract each $\mathbf{z}_t$, which are then later nudged during the reverse process using

$$\mathbf{z}_t^{edited} = \mathbf{z}_t + \nabla L \cdot \alpha_t^2. \tag{19}$$

where ∇L is derived from $\mathbf{P}_t$. As far as I know, I am one of the first to use a binary attribute classifier to nudge each $\mathbf{z}_t$ in this way. For the DDIM I use 1000 inference steps and for the DDPM I use 100. I make use of random images from the Flickr8k data set [6] and one of myself. Each image is aligned to a standard pose or orientation using the alignment script from [33] before being edited.

From figure 18, it can be found that attributes such as smiling, no beard, big nose, bags under eyes and youth can be edited. However, the edits are minuscule and not always necessarily realistic. Next, I invert an image of myself using equation 7 and edit it using each gradient function. Figure 17 illustrates the results. It shows that the binary classifier attribute, smiling, can affect real images (left image). It also demonstrates that the segmentation network can be applied to change the nose. Further, colour edits can be easily applied without much finetuning of the parameters λ and $[t_1, t_2]$.

The main challenge of using the DDIM inversion technique is the requirement of 1000 time steps. However, another problem is the latent $\mathbf{x}_T$ not being a true Gaussian, this has also been explored before by [23], see appendix B for further details. All things considered, the DDPM inversion method is preferred as it results

Figure 17: **Real image edits with DDIM inversion.** Left: Edit using the binary classifier with the negative score for *smiling* as the loss, with masking of gradients computed using the image segments; mouth, upper lip and lower lip. Middle: Segmentation network edit of the nose and no masking of the gradients. Right: Colour guidance of the hat.

Figure 18: **Edits of real images using DDPM inversion.** Each image is edited using the binary classifier. Model: DDPM with 100 inference steps and DDPM inversion. Top row (Tskip=95, guidance strength=1): Original, Added beard, arched eyebrows, bags under eyes, age. Middle row (Tskip=95, guidance strength=50): Original, smile, bigger nose, bags under eyes, age. Last row (Tskip=0, guidance strength=90, t1=0, t2=80): Original, smile, bigger nose, bags under eyes and age.

in more realistic edits and can be applied much faster. However, this is only a qualitative experiment and the findings are based on a limited number of examples.

5 Limitations

This thesis is limited in the scale and number of quantitative measures due to the computational requirements of generating images from DDMs and the limited computational resources available.

6 Discussion and Conclusion

In conclusion, I have investigated the use of gradient guidance methods for editing images using pretrained DDMs. I have presented three universal guidance methods, specifically, colour guidance, segmentation network guidance and binary attribute classifier guidance, and demonstrated their effect qualitatively, showing successful edits on synthetic images. However, I have also outlined the challenges related to the overall approach, with one being the choice of the hyperparameters. I have addressed these challenges by analysing the effect of the hyperparameters in more detail and arrived at the same conclusions as previous studies. Furthermore, I have presented a regularization method for entangling and disentangling attributes of the binary classifier, showing promising results. Additionally, I have explored the use of the binary attribute classifier for editing the *age* attribute and found it to work as intended with some changes to correlated attributes. Finally, I have edited real images using the binary attribute classifier and found the edits to be minuscule and less convincing than synthetic image edits, further indicating room for improvement in future work.

7 Future Work

While I have demonstrated the quantitative effect of the binary classifier for the attribute *age*, using attribute consistency, other attributes and DDMs could be analysed similarly to verify their applicability.

Furthermore, the results of real-image editing showed that both the binary classifier and segmentation model can be improved as the edits were either tiny or non-existent.

Additionally, research into methods that evaluate the image quality and faithfulness to the original and guidance function would be interesting.

Lastly, the use of other masking methods that allow small changes to the areas outside $\mathbf{M} = 1$ is needed. One possible method is $L = L(\mathbf{P}_t \odot \mathbf{M}) + \lambda LPIPS((1 - \mathbf{M}) \odot \mathbf{P}_t, \mathbf{x}_0)$. This allows changes to the complementary area of the mask, controlled by the hyperparameter λ and the metric LPIPS.

Acknowledgements

This thesis would not have been the same without the great support and guidance from my two supervisors; Stella Grasshof and René Haas.

I would also like to show appreciation for the loving support my family has given me during this process.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[4] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14367–14376, October 2021.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[6] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 15–29. Springer, 2010.

[7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. 2022.

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

[9] René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models. *arXiv preprint arXiv:2303.11073*, 2023.

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. 2022.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[13] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv preprint arXiv:2304.06140*, 2023.

[14] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[15] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022.

[16] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

[17] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5548–5557, 2019.

[18] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! image synthesis with semantic diffusion guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[21] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[22] Alex 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. *arXiv preprint arXiv:2112.10741*, 2021.

[23] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decoda½ble representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[25] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. *URL https://arxiv. org/abs/2204.06125*, 7, 2022.

[26] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[28] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[31] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020.

[33] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *arXiv preprint arXiv:2102.02766*, 2021.

[34] Dani Valevski, Matan Kalman, Eyal Molad, Eyal Segalis, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning a diffusion model on a single image. *ACM Trans. Graph.*, 42(4), jul 2023.

[35] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[36] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

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

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