Improved Zero-Shot Image Editing via Null-Toon and Directed Delta Denoising Score

Improved Zero-Shot Image Editing via Null-Toon and Directed Delta Denoising Score

Masud An Nur Islam Fahim[1 [0000−0002−0295−5965]] and Jani Boutellier[1 [0000−0001−7606−3655]]

University of Vaasa, Vaasa, Finland
{masud.fahim, jani.boutellier}@uwasa.fi

Abstract. Recently, there has been a rapid surge in the utilization of diffusion models for customized image generation and editing tasks, especially using zero-shot editing algorithms that can largely operate on given images regardless of their source domain. This work is based on two well-known zero-shot image editing algorithms: Null Text Inversion (NTI) and Delta Denoising Score (DDS). With respect to NTI, we mainly focus on image cartoonization, which has received less attention in the context of text-guided image editing. In a nutshell, we propose a customized reconstruction phase for NTI, which helps transforming the natural input image into cartoon images with desired customization by supporting parameters. We also improve the current DDS optimization baseline and propose the Directed Delta Denoising Score (DDDS). Our DDDS algorithm offers a better image editing experience by replacing the target text prompt with the proposed *directed text prompt*. Computing directed text prompt requires one subtraction operation and yields significant reconstruction improvement over DDS. To demonstrate the effectiveness of our contributions, the paper presents both quantitative and qualitative comparisons against the state-of-the-art, as well as several visual examples.

Keywords: Diffusion model · Zero-shot editing · Image generation.

1 Introduction

Diffusion models [21, 10, 18, 20] have shown great promise in text-guided image editing, especially with natural image editing tasks like image inpainting [4, 13, 14, 25], style transfer [8, 22, 2], text-guided editing [8, 2, 15, 24, 7, 11], or segmentation [1, 3]. Our study is particularly interested in text-guided image editing of natural images, where we want to edit natural images based on prompt input.

Editing natural images with a diffusion model is not straightforward. Typically, zero-shot editing studies rely on inversion strategies like DDIM [21] to access the desired latent space of the inputs. The choice of inversion strategy is critical for reconstruction fidelity. Null Text Inversion (NTI) [15] is a popular method for image editing that relies on Denoising Diffusion Implicit Models (DDIM) inversion. In this study we have observed that using NTI it is possible to transform any natural image into a cartoonified appearance by modifying the reconstruction path leveraging predicted noise via weighted augmentation, and optimized null text perturbation. Additionally, parameterized noise perturbation is considered to increase or decrease the desired degree of

(a) Input image (b) NTI output [15] (c) Our output

Fig. 1: Here, output quality comparison between the Null Text Inversion (NTI) [15] and the proposed cartoon reconstruction algorithm: the proposed algorithm's output is free from text artifacts (image center) in contrast to the NTI output. Additionally, our algorithm cartoonifies the whole image, while [15] focuses on the dog region only.

(a) Input image (b) NTI output [15] (c) Our output

Fig. 2: Image editing comparison between the delta denoising score (DDS) [7] and the proposed directed delta denoising score (DDDS). Here, the source prompt is *Two cats sitting by the mirror*, and the target prompt is *Two* ***origami*** *cats sitting by the mirror*. DDS edited regions are blurry and missing salient details. In contrast, our DDDS offers crisper output, including reflection area.

output image detail. Since our approach stands upon the null text inversion approach and is dedicated to the cartoon transformation of the natural images, we name our approach *Null-Toon*. A recent study [26] also showed that it is possible to translate natural images into cartoons using null text optimization. However, in contrast to the proposed work, their process does not offer parameterized tunability. Moreover, our algorithm returns cartoon images without any traces of text artifacts, as shown in Figure 1.

In addition to the particular cartoon reconstruction approach, we complementarily propose another method for superior general-purpose zero-shot image editing. Delta Denoising Score (DDS) [7] is an efficient alternative to the inversion-dependent methods [15, 11, 23]. DDS harnesses the core idea of Score Distillation Sampling (SDS) [19] and improves over SDS. In practice, SDS returns noisy gradients during optimization, resulting in unsatisfactory editing performance. DDS rectifies SDS by taking in two sets of latent-text pairs and performing SDS optimization for individual pairs, subtracting the estimated gradients from each pair [7]. This novel *delta* gradient obtained from subtraction is cleaner and contains indicates direction for editing.

However, reconstructed images from this *delta* gradient often contain traces of the source image, resulting in unfavorable edits [7]. We rectify this situation by proposing *directing* the target prompt via weighted subtraction. This elementary change helps in

drastic reconstruction performance over DDS, naming our method as Directing Delta Denoising Score (DDDS). Our DDDS approach only needs an extra subtraction operation as an additional computation cost to DDS, and for the same optimization configuration, editing performance is superior over DDS. We show the necessary visual comparison between our study and other studies in the later section of the manuscript and summarize overall contributions as follows:

- An algorithm for cartoonifying natural images using null-text reconstruction,
- Providing user-adjustability to algorithms output via parametrized noise perturbation and
- Proposing the novel Directed Delta Denoising Score (DDDS) for crisper output, compared to the existing Delta Denoising Score (DDS).

2 Related Work

Text-to-image models [21, 10] have shown great promise in image generation conditioned by an input text prompt. These works harness powerful diffusion models for generation or guided reconstruction tasks. Recently, we have seen a surge of zero-shot image editing approaches where customized algorithms provide general-purpose image editing applications without tuning large diffusion models. These image editing can roughly be divided into the following clusters: end-to-end editing [12, 18, 5], attention manipulation [8, 2, 6, 16], and DDIM inversion families [15, 11, 24]. Score-based editing approaches [19, 7, 17] are an efficient alternative to inversion-dependent studies, where image editing is done end-to-end. Our study is related to both score and inversion-based studies.

3 Proposed Algorithms

We begin by revisiting the basic concepts regarding the diffusion model: we represent the dataset as $\boldsymbol{\mathcal{D}}$, clean image or latent as $\boldsymbol{z_0}$, text embedding (source) as $\boldsymbol{y}$, noise as ϵ, T as time step, and the diffusion model as ϵ_θ. Typically, the following objective is used to describe the training of a diffusion model:

$$\min_{\theta} \mathbb{E}_{\mathbf{z}_0 \sim \mathcal{D}, \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}), \mathbf{t} \sim U(1,T)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, \mathbf{y}, \mathbf{t} \right) \right\|^2$$

In the above, $\mathcal{D}$ denotes the utilized dataset, $\mathbf{z}_t$ is a noisy iteration of the image $\mathbf{z}_0$ at timestep $\mathbf{t}$, and $\mathbf{y}$ is a corresponding conditional embedding. After model training, we can sample a new image given condition $\mathbf{c}$ with the commonly used DDIM sampler,

$$\begin{aligned} \mathbf{z}_{t-1} &= \mathrm{DDIM} \left(\mathbf{z}_t, \boldsymbol{\epsilon}_t, \mathbf{t} \right) \\ &= \sqrt{\alpha_{t-1}} \cdot f_\theta \left(\mathbf{z}_t, \mathbf{c}, t \right) + \sqrt{1 - \alpha_{t-1}} \cdot \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, \mathbf{y}, t \right), \end{aligned}$$

where $f_\theta \left(\mathbf{z}_t, \mathbf{y}, \mathbf{t} \right) = \frac{\mathbf{z}_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta (\mathbf{z}_t, \mathbf{y}, \mathbf{t})}{\sqrt{\alpha_t}}$, $\boldsymbol{\epsilon}_t = \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, \mathbf{y}, \mathbf{t} \right)$, and α_t is a time-varying noise parameter in DDIM. For our study, we consider the above formula as the DDIM sampler.

Algorithm 1: Image reconstruction from Null text trajectory

Input: Noisy latent $\mathbf{z}_T$, Null text trajectory $\mathcal{Y}_\phi$, target text $\hat{\mathbf{y}}$, balance parameter ω
Output: Edited latent $\mathbf{z}_0^*$

Set $\mathbf{z}_t = \mathbf{z}_T$;
for $t \leftarrow T$ **to** 0, $i \leftarrow 1$ **to** $len(\mathcal{Y}_\phi)$ **do**
  $\mathbf{y}_t^\phi = \mathcal{Y}_\phi[i]$
  $\epsilon_t \leftarrow \mathcal{CFG}(\mathbf{z}_t, \mathbf{y}_t^\phi, \hat{\mathbf{y}}, \mathbf{t}, \omega)$, Eqn. 1
  $\mathbf{z}_{t-1}^* \leftarrow DDIM(\mathbf{z}_t, \epsilon_t, \mathbf{t})$
end
Return adapted latent $\mathbf{z}_0^*$

Now, if we want to apply the diffusion model for local or global edits, it is necessary to invert them into $\mathbf{z}_0 \rightarrow \mathbf{z}_T$, DDIM inversion [21] being a popular way to do this. After having $\mathbf{z}_T$, we can expect a lossless return to $\mathbf{z}_0$ with perfect reconstruction, which is unfortunately not possible in reality. To address this, null text optimization [15] has been proposed as a highly efficient way to recover $\mathbf{z}_0^*$, which is very close to the original $\mathbf{z}_0$. Once we have near-perfect reconstruction approach, the influence from the target prompt or text embedding $\hat{\boldsymbol{y}}$ can be integrated to reconstruct the input image with the desired edits using optimized null text embeddings and $\hat{\boldsymbol{y}}$ via classifier-free guidance [9].

By design, null text optimization returns a trajectory of unconditional text embeddings $\mathcal{Y}_\phi = [\boldsymbol{y}_{\boldsymbol{T}_\phi}, ..., \boldsymbol{y}_{\mathbf{0}_\phi}]$, and by utilizing them for any time step $\mathbf{t}$, classifier-free guidance can be expressed as follows:

$$\epsilon_t^* = \epsilon_\theta(\mathbf{z}_t, \mathbf{t}, \mathbf{y}_t^\phi) + \omega * (\epsilon_\theta(\mathbf{z}_t, \mathbf{t}, \hat{\mathbf{y}}) - \epsilon_\theta(\mathbf{z}_t, \mathbf{t}, \mathbf{y}_t^\phi)) \tag{1}$$

Here, ω is the impact factor for balancing the information contents from the source and target content. By leveraging the above concepts, we can denote the prompt-guided reconstruction steps as follows (Algorithm 1):

Using Algorithm 1, we can obtain $\mathbf{z}_0^*$ that contains the natural image with respect to the target prompt $\hat{\mathbf{y}}$.

3.1 Null-Toon

Having described the necessary concepts above, we can proceed with a detailed description of the proposed Null-Toon algorithm based on 1. After completing null text optimization, we get the optimized null text trajectory $\mathcal{Y}_\phi$, and the length of the trajectory is $\mathcal{Q}^\phi$. Based upon this trajectory with the help of Algorithm 1, we first reconstruct the original image while setting $\hat{\mathbf{y}}$ as an empty statement to introduce no change in the reconstruction. This self-reconstruction also allows us to collect the predicted noise trajectory $\mathcal{E}$, containing the output noise $\mathbf{y}_t^\phi$ from the classifier free guidance step. Apart from $\mathbf{z}_T$, and $\mathcal{Y}_\phi$, we consider $\mathcal{E}$ as another input to our toonification algorithm along with a few other parameters as coefficients and threshold conditions. We formulate our toonification approach in Algorithm 2.

Algorithm 2: Cartoonifying from Null text trajectory

Input: Noisy latent $\mathbf{z}_T$, Null text trajectory $\mathcal{Y}_\phi$, target text $\hat{\mathbf{y}}$, Predicted noise trajectory $\mathcal{E}$, trajectory length $\mathcal{Q}^\phi$, Convex weights $\mathcal{A}$, balance parameter ω. time-steps t_1, t_2, coefficients c_1, c_2, c_3, c_4, c_5

Output: Edited latent $\mathbf{z}_0^*$

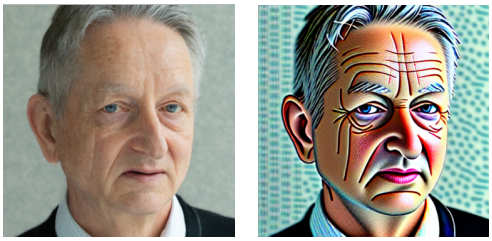
1 Set $\mathbf{z}_t = \mathbf{z}_T$;
2 **for** $\mathbf{t} \leftarrow T$ **to** 0, $i \leftarrow 1$ **to** $\mathcal{Q}^\phi$ **do**
3 Set $w_1 = \mathcal{A}[i]$ & $w_2 = 1 - w_1$;
4 $\mathcal{Q}^\phi$ - i ;
5 $\mathbf{y}_t^\phi = \mathcal{Y}_\phi[i]$
6 **if** $\mathbf{t} < t_1$ **then**
7 $\mathbf{y}_t^\phi = \mathbf{y}_t^\phi + c_1 * \mathcal{Y}_\phi$ [$\mathcal{Q}^\phi$ - i]
8 **end**
9 $\epsilon_t \leftarrow \mathcal{CFG}(\mathbf{z}_t, \mathbf{y}_t^\phi, \hat{\mathbf{y}}, \mathbf{t}, \omega)$, Eqn. 1
10 **if** $\mathbf{t} > t_2$ **then**
11 $\epsilon_t = \epsilon_t + c_2 * \mathbf{z}_t$
12 **end**
13 $\epsilon_t = w_1 * \mathcal{E}[i] + w_1 * \epsilon_t$
14 $\epsilon_t = ConvexSum(\mathcal{E}[i], \epsilon_t, c_3)$
15 $\epsilon_t^s = Enhance(\epsilon_t, c_4)$ **| Enhance sharpness |**
16 $\epsilon_t^d = Enhance(\epsilon_t, c_5)$ **| Enhance smoothness |**
17 $\epsilon_t = ConvexSum(\epsilon_t^d, \epsilon_t^s, c_3)$
18 $\mathbf{z}_{t-1}^* \leftarrow DDIM(\mathbf{z}_t, \epsilon_t, \mathbf{t})$
19 **end**
20 Return adapted latent $\mathbf{z}_0^*$

Similar to Algorithm 1, we start from anchoring the latent $\mathbf{z}_T$, and setting $\mathbf{z}_t = \mathbf{z}_T$ to return back to $\mathbf{z}_0$. Based upon on Algorithm 1, we propose Algorithm 2. Below, we explain the impact of specific pseudocode lines with visual examples.

Augmenting null text (Alg. 2 line 6) with *future* optimized null text brings in the main cartoonifying impact through drastic loss in image detail, as shown in Figure 3. In our algorithm, *future* represents the optimized null texts from later stages of the iteration; we already have access to the null text trajectory and access those texts in reverse. If we use Algorithm 1 for reconstruction and augment the null text for a given time step by recalling the null text from the future time step, we observe such cartoon style reconstruction. However, we impose a threshold for the time step to preserve the image details. We obtained the optimal value for the timestep thresholds via trialing on multiple images.

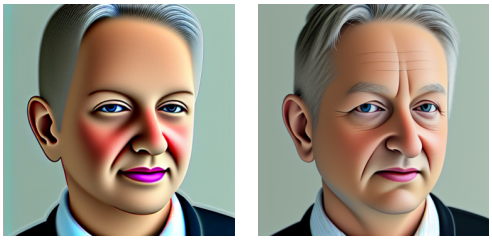
As the augmentation procedure provides the cartoon effect and loss of detail, an appropriate rectification is required to recover detail again, which is acquired through later steps of the proposed Null-Toon algorithm 2.

Augmenting latent (Alg. 2 line 10) with the predicted noise $\epsilon_\mathbf{t}$ helps reducing loss of detail while preserving comic-like appearance. Augmenting the estimated latent $\mathbf{z}_t$ with $\epsilon_\mathbf{t}$ provides reconstruction as depicted in Figure 4. From Figure 4, we can see that with the dominance of the smoothing effect due to latent augmentation, content details



(a) Input image (b) Output image

Fig. 3: Impact of augmenting *future* null text with Algorithm 1.



(a) Impact of augmenting $\mathbf{z}_t$ to the $\epsilon_{\mathbf{t}}$. (b) Impact of augmenting $\mathcal{E}[i]$ to the $\epsilon_{\mathbf{t}}$.

Fig. 4: Impact of augmenting $\mathbf{z}_t$ and $\mathcal{E}[i]$ during cartoon reconstruction.

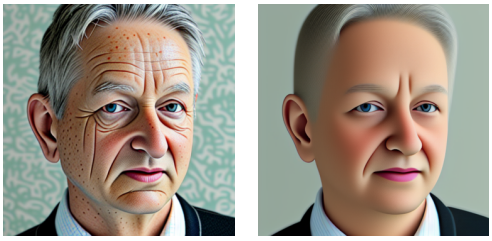
have disappeared significantly. Optimally, we would like to balance between the two extreme conditions while preserving the cartoonish appearance.

Augmenting noise from null-text optimization (Alg. 2 lines 12-13, 16) with the predicted noise $\epsilon_{\mathbf{t}}$ helps in regaining more image details. From Algorithm 1, null-text reconstruction relies on optimized null text trajectory only [15]. Another study [24] showed that null text optimization returns a trajectory of noise $\mathcal{E}$ that contains predicted noise $\epsilon_{\mathbf{t}}$ while $\mathbf{z}_0 \rightarrow \mathbf{z}_T$, and it is possible to utilize $\mathcal{E}$ in Algorithm 1 after the CFG step to gain more context preserving reconstruction [24] while reverting $\mathbf{z}_T \rightarrow \mathbf{z}_0$.

We have adopted this idea to integrate more details into our reconstruction. Following [24], we get a series of convex weights $\mathcal{A}$, tailored for each iteration of the diffusion step, and based upon these weights, we perform the convex sum between the predicted noises from $\mathcal{E}$, and current estimation of $\epsilon_{\mathbf{t}}$. To reinforce the details, we perform another extra convex summation step as present in the Algorithm 2. Consequently, as shown in Figure 4, our current reconstruction is layered with a desirable cartoon look and slightly lacking realistic details.

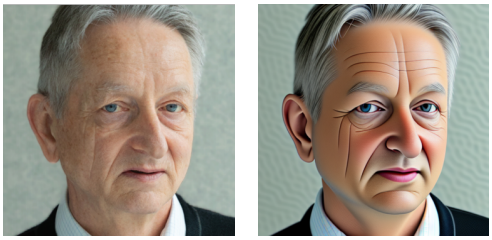
Perturbing the predicted noise (Alg. 2 lines 14-15) $\epsilon_{\mathbf{t}}$ helps in emphasizing smoothness or details if we apply our proposed mean-directed perturbation. In summary, our mean-directed perturbation is done by following steps:

- Say $\epsilon_{\mathbf{t}}$ consists of $\mathbf{n}$ channels, and we estimate the mean $\bar{\epsilon}_{\mathbf{t}}$ across all channels.
- We add noise to $\mathbf{n}$ instances of $\bar{\epsilon}_{\mathbf{t}}$, and concatenate them together as $\hat{\epsilon}_{\mathbf{t}}$, obtaining the same shape as $\epsilon_{\mathbf{t}}$.



(a) Detail-rich output (b) Low detail output

Fig. 5: Impact of the convex weights: $\mathbf{c_4} = 0.96$ causes more details, whereas $\mathbf{c_5} = 1.01$ causes overtly smooth output.



(a) Input image (b) Output image

Fig. 6: Final reconstruction from the proposed Algorithm 2.

– Performing a convex sum between $\epsilon_\mathbf{t}$ and $\hat{\epsilon}_\mathbf{t}$ results in less or more details, depending on the applied coefficient, as shown in Figure 5.

Figure 5 highlights that manipulation of $\epsilon_\mathbf{t}$ via $\mathbf{c_4}$ and $\mathbf{c_5}$ affects the level of detail in the output. Merging both together through a final convex summation provides the desired $\epsilon_\mathbf{t}$, which balances the degree of detail, smoothness, and cartoon effect together. Performing DDIM sampling upon the final sum until the last iteration returns the desired cartoon image as present in Figure 6. In our algorithm, Convex sum means regular convex summation operation.

Our proposed Null-Toon uses several hyperparameters to achieve the desired outcome. Most of the hyperparameters are empirically found and need only minimal tuning during reconstruction. Among those parameters, $\mathbf{c1}, \mathbf{c2}$ and $\mathbf{c3}$ are fixed to values $\mathbf{4.5}, \mathbf{3.5}, \mathbf{0.5}$, respectively. $\mathbf{c4}$ and $\mathbf{c5}$ remain adjustable to balance between cartoonish appearance and image details. In our experiments, $\mathbf{c4}$ ranged between $[0.95, 0.98]$, and $\mathbf{c5}$ between $[1.02, 1.04]$. For the convex weights $\mathcal{A}$, we followed the weight distribution process of [24]. We kept $t_1 = 220$ and $t_2 = 920$, which was empirically obtained first via trialing on several natural images. and later set as a constant during the test.

3.2 Directed Delta Denoising Score (DDDS)

Starting from DDS [7], we can write the image editing Equation 2 for Score Distillation Sampling (SDS) as follows:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \epsilon, \mathbf{t}) = \epsilon_\theta^\omega \left((\hat{\mathbf{z}}_\mathbf{t}, \mathbf{t}) - \epsilon\right) \frac{\partial \hat{\mathbf{z}}_\mathbf{t}}{\partial \theta} \tag{2}$$

However, editing with the formulation above results in a blurry output [7], often flattening out the context along with the regions, showing alignment towards the text prompt $\mathbf{y}$. The rationale for the above situation is more understandable via the following decomposition:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \epsilon, \mathbf{t}) := \delta_{text} + \delta_{bias} \tag{3}$$

According to DDS [7], δ_{text} makes the effective gradient drift towards the desirable direction that aligns with the prompt, and δ_{bias} conditions the outcome towards undesirable directions. The authors of DDS [7] provide an interesting solution to the above Eq. 3 by

$$\nabla_\theta \mathcal{L}_{\text{DDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \epsilon, \mathbf{t}) = \nabla_\theta \mathcal{L}_{\text{SDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}) - \nabla_\theta \mathcal{L}_{\text{SDS}}(\mathbf{z}, \mathbf{y}). \tag{4}$$

DDS [7] showed that $\nabla_\theta \mathcal{L}_{\text{DDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}, \epsilon, \mathbf{t})$ is almost equivalent to δ_{text}, because Eq. 4 returns a cleaner gradient by performing the proposed subtraction. In practice, Equation 4 is effective, because in each iteration it estimates two directions dedicated to the target text and source text. Subtraction between two of them returns a unique direction that contains the necessary edits from the $\nabla_\theta \mathcal{L}_{\text{SDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}})$, and additional context bias from $\nabla_\theta \mathcal{L}_{\text{SDS}}(\mathbf{z}, y)$.

However, Eq. 4 fails to capture the desired edits in many cases, and the reason is apparent from the formulation of DDS. In Eq. 4, the editing direction is obtained from the gradient, and the gradient stems from the estimated noise difference between $\hat{\mathbf{z}}, \hat{\mathbf{y}}$ and $\mathbf{z}, \mathbf{y}$ pairs. For any given time step $\mathbf{t}$, from $\hat{\mathbf{z}}, \hat{\mathbf{y}}$ we get $\hat{\epsilon}_\mathbf{t}$, and ϵ_t from the $\mathbf{z}, \mathbf{y}$ pair. Here, we argue that if the difference between $\hat{\epsilon}_\mathbf{t}$, and ϵ_t is more biased towards the desired edit and minimal towards the context from the original image, then we can capture refined details that match $\hat{\mathbf{y}}$.

We can only emphasize the target prompt in the DDS setup by tuning ω in the classifier-free guidance stage. As a result, the DDS gradient is trimmed by the influence from ϵ_t, preserves the context, limits the edit fidelity, and improves the fundamental lack of image details present in SDS. This gives us two objectives to satisfy, which are listed below, along with proposed solutions:

- *How can $\hat{\mathbf{y}}$ be emphasized more?* We address this question with our proposed subtraction operation, where we direct the target prompt $\hat{\mathbf{y}}$ by $\hat{\mathbf{y}}_\mathbf{d} = \alpha_1 * \hat{\mathbf{y}} - \alpha_2 * \mathbf{y}$. Here, α_1 typically ranges between [1.1, 1.3], and $\alpha_2 = 0.05$. By this, we obtain a *directed* prompt that is more biased towards the intended image edits.
- *How can the effect of $\mathbf{y}$ be reduced?* In both SDS and DDS, $\mathbf{y}$ steers the gradient direction towards the source context, preserving the source image's similarity where editing is unnecessary. However, $\mathbf{y}$ reduces detail in the edited image in SDS, but

results in undesirable results in DDS. We minimize this situation by considering an auxiliary null-text embedding $\mathbf{y_n} = \Phi$. In other words, we have used $\mathbf{y_n}$ instead of $\mathbf{y}$ in the original DDS equation. As a result, in the optimization phase, $\mathbf{y}$ puts least impact on the estimated gradients without causing undesirable alteration.

To summarize, our directed delta denoising score (DDDS) returns a cleaner gradient than DDS by projecting $\hat{\mathbf{y}}$ to $\hat{\mathbf{y}}_\mathbf{d}$, where editing embedding gets more attention. Additionally, to reduce the over-emphasizing effect of source embedding $\mathbf{y}$, we replace it with a null-text embedding $\mathbf{y_n}$, and $\mathbf{y_n} = \Phi$. Hence, the proposed directed delta denoising score is as follows:

$$\nabla_\theta \mathcal{L}_{\text{DDDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}_\mathbf{d}, \epsilon, \mathbf{t}) = \nabla_\theta \mathcal{L}_{\text{SDS}}(\hat{\mathbf{z}}, \hat{\mathbf{y}}_\mathbf{d}) - \nabla_\theta \mathcal{L}_{\text{SDS}}(\mathbf{z}, \mathbf{\Phi}) \tag{5}$$

In Eq. 5 estimated gradients are more precise than the gradients of Eq. 4 due to the above-mentioned reasons. We present the necessary demonstration in the following sections to support our claim.

4 Results

This section presents a visual comparison between the proposed methods and previous studies. We have listed separate baseline methods for each technique to present a fairground for the comparisons. Our Null-Toon algorithm uses null text inversion as its core process; we select the Null Text Inveriosn (NTI)[15], Direct Inversion (DI) [11], and Null Text Guidance (NTG) [26] to compare the cartoon translation performance. These three works use null text optimization as the central part of the given zero-shot image editing task. Similarly, we compare our directed delta denoising score algorithm with two other score-dependent algorithms: Delta Denoising Score (DDS) [7] and Contrastive Denoising Score (CDS) [17].

4.1 Implementation details

For all the comparisons with baselines, we refer to the official code repositories of the respective studies. In the case of Null-Toon, we have not changed hyperparameters for other studies. However, score-based algorithms [17, 7] depend heavily on the iteration count. Given the complexity of the image and corresponding target prompt, the iteration count can vary from case to case. Our experiments showed that a higher iteration count results in over-editing and vice versa; pinpointing the ideal iteration count is not viable either. To present the results, we run DDS [7], CDS [17], and the proposed DDDS for 350 iterations.

4.2 Editing comparison

Figure 7 demonstrates the visual comparison between the proposed cartoon translation algorithm and other baselines [26, 15, 11]. As shown in the topmost row of Figure 7, our translation algorithm returns a crisper cartoon edit, whereas NTG [26] returns blurry output, and the other algorithms [15, 11] perform almost no change to the input.

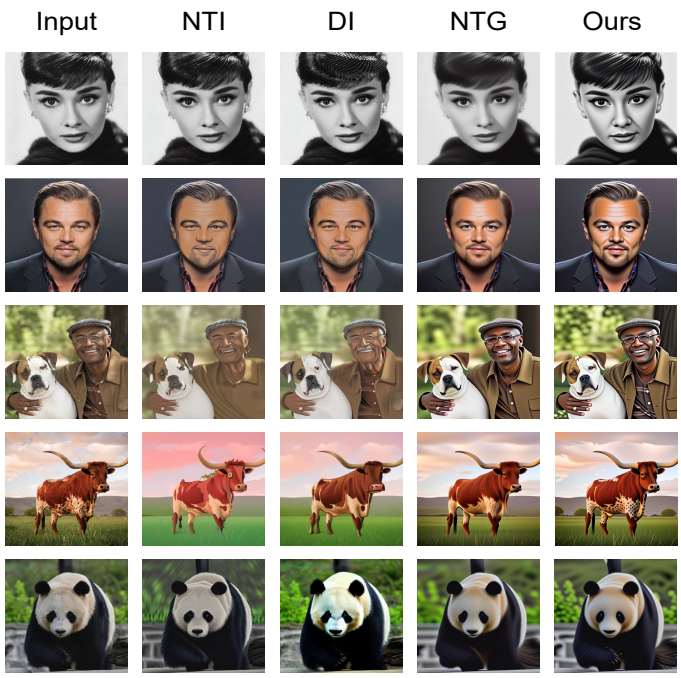


Fig. 7: Comparison of cartoon reconstruction between NTI [15], DI [11], NTG[26], and ours stands for the proposed Null-Toon algorithm.

In the second row, all of the baselines [26, 15, 11] provide reasonable editing results, but NTI[15] and DI [11] slightly alter the personal appearance of the character. Exaggerating the character is prevalent with NTI [15] and DI [11]; clearly, the character has become older, whereas NTI [15], and NTG [26] alter the dress of the character as well. Compared to these, our algorithm does not emphasize the sensitive features of the characters, which is evident from the given examples.

This trend is consistent with the rest of the examples as well, where our algorithm does not bend the horn of the cow or loosely reconstruct the leg of the panda while translating the images into cartoons. We can conclude that our cartoon algorithm can cartoonify natural photos without altering the salient features.

From Figure 8, we can see the editing result comparison between the proposed directed delta denoising score (DDDS) algorithm and other baselines [7, 17]. In the first row, our algorithm and [17] have returned the respective reconstruction of the cat image into anime art, unlike [7], which distorts the cat appearance in its reconstruction. In the second row, the task is to transform the lion's head into an origami structure. Surprisingly, both DDS [7] and CDS [17] return the output with minimal deviation from the input. On the contrary, our approach almost translates the lion's head into an origami-looking head. In the next row, DDS [7] transforms the chicken's beak with a metal look, and CDS [17] brings an extra chicken head on the top of the tail of the other chicken. Our method does not return such unnatural edits, but it does change the color

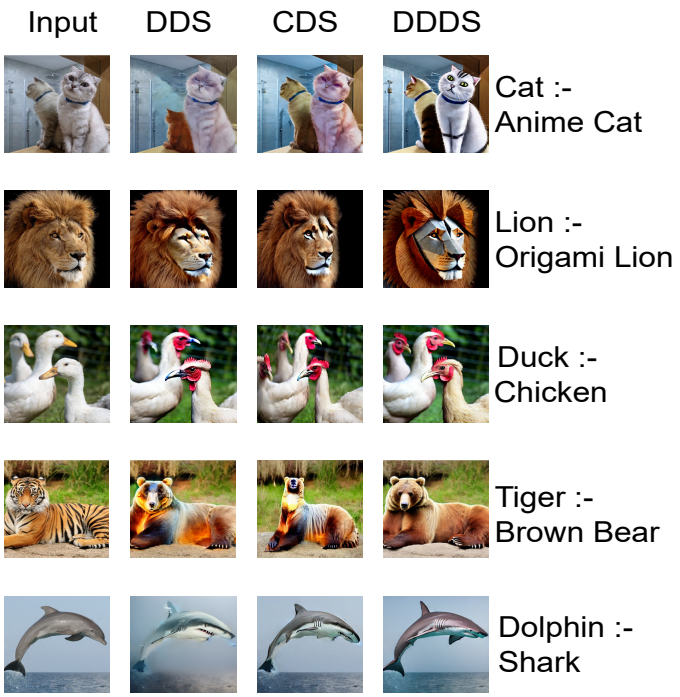


Fig. 8: Here, we show the zero-shot editing comparison between DDS [7]. CDS [17], and our DDDS. From the visual appearance, our algorithm offers better editing fidelity and reconstruction that closely matches the target prompt.

of one chicken into a yellowish look, which is an anomaly compared to the all-white ducks from the input.

Likewise, our method remains consistent with its output quality with the rest of the images; for example, our algorithm transforms the tiger into a bear without distorting its head like CDS [17] does, or dolphin to a shark without changing the context like [7]. From this comparison, our algorithm brings significant improvement and consistency with zero-shot image editing tasks using score-based methods.

4.3 Impact of c_4

In Figures 9 and 10, we show the impact of hyperparameter c_4 on our proposed cartoon reconstruction algorithm. We can see that lowering c_4 leads to higher image details for both images. Especially in Figure 9, it can be seen that increasing c_4 greatly affects the amount of detail in the background, simultaneously resulting in a sharper cartoon image of the tiger. In Figure 10, lowering the value of c_4 makes the edges of the face become sharper without losing the baseline cartoon look. Although our cartoon reconstruction algorithm has several other hyperparameters as well, we kept these fixed during the ablation study. Due to space limitations, it is not possible to explore the effects of other hyperparameters.

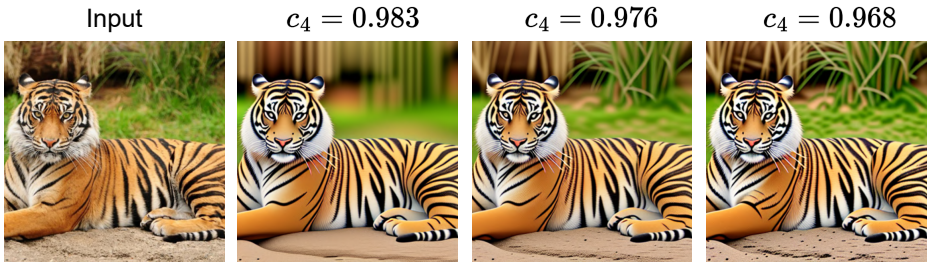


Fig. 9: In this picture, we show the impact of parameter c_4 on controlling image detail while translating the input image into a cartoon image using our Null-Toon algorithm. It is evident that lowering c_4 preserves the cartoon image closer to the original.

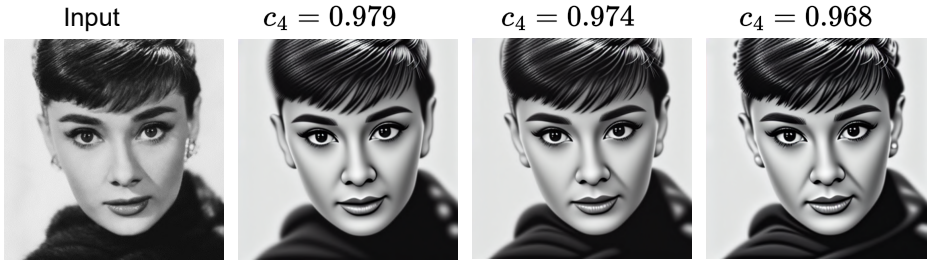


Fig. 10: The impact of parameter c_4 on a human face: by reducing c_4, we can bring more detail to the character's face.

4.4 Local edits using DDDS

In Figure 11, we show the performance comparison for editing local regions between delta denoising score [7] and our study. Here, we take an image of a dog and add men's accessories to the dog, which were not present in the original image. For the sunglasses, we see that our algorithm can return an edited image where the added sunglasses appear more natural than with DDS [7]. Similarly, while adding a hat on the top of the dog's head, our algorithm can edit the region without distorting the original structure of the dog. On the contrary, one eye of the dog is undesirably edited while adding hat by DDS [7]. Finally, the dog's tie is semi-reconstructed, and the dog's color is drastically shifted by DDS [7] while adding a suit to the dog's image. Compared to that, editing results from our algorithm are more crisp and character-preserving.

Algorithm	CLIP similarity	LPIPS
DDS [7]	32.19	$0.14 \pm .07$
CDS [17]	33.06	$0.14 \pm .02$
DDDS	**33.79**	$\mathbf{0.13 \pm .04}$

Table 1: CLIP-similarity and LPIPS comparison between DDS

[7], CDS [17], and the proposed DDDS. For CLIP, higher is better and for LPIPS lower is better.

Fig. 11: Here we demonstrate the editing difference between our algorithm and the baseline DDS [7] approach. Our algorithm achieved better visual reconstruction performance in all cases while keeping minimal context distortion.

4.5 After edit text image consistency

To evaluate the editing performance of our proposed DDDS algorithm, we used text-image similarity evaluation after editing completion. For this evaluation, we have used CLIP similarity, and this approach is adopted in other zero-shot editing studies as well [11, 19]. In the CLIP similarity evaluation, we take in the edited image and its corresponding prompt and then calculate their similarity by projecting the image and text into the same embedding space. However, we can obtain this evaluation in two ways: estimating the similarity score for the edited region only by using a human-annotated mask or the whole image where we do not mask the non-edited regions. For our study, we have estimated the CLIP similarity score for the entire image, as presented in Table 1. Note that the CLIP similarity score heavily relies on the rich textual description, and depending on the text prompt, we observed variation within the estimated scores. For this evaluation, we have compared our DDDS algorithm with DDS [7] and [17]. From the table, the obtained similarity score from our DDDS algorithm outperforms other methods for the edited images. We also include learned perceptual image patch similarity (LPIPS) score by following [17, 7]. As shown in Table 1, the proposed work also achieves satisfactory performance using LPIPS metric.

5 Conclusion

In our study, we proposed two different methods for zero-shot image editing. Our first algorithm is dedicated to image cartoonization. We devised a unique reconstruction process by leveraging the null text trajectory to revise the previous noisy latent to introduce the cartoonization effect. In our reconstruction phase, we enabled a novel tuning setup for the user to include desired control over image details or smoothness without tampering the content. In our second algorithm, we show that *directing* the target text prompt via weighted subtraction between the source and target text prompt. As a result, our DDDS optimization returns more *directed* gradients and results in a more realistic

image editing result without introducing zero computation increase over the original DDS optimization algorithm. We plan to extend our work to video and 3D instances in the future.

5.1 Acknowledgements

This work was partially funded by the ERDF project AI2Business.

References

1. Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. arXiv preprint arXiv:2112.03126 (2021)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
3. Burgert, R., Ranasinghe, K., Li, X., Ryoo, M.S.: Peekaboo: Text to image diffusion models are zero-shot segmentors. arXiv preprint arXiv:2211.13224 (2022)
4. Choi, J., Kim, S., Jeong, Y., Gwon, Y., Yoon, S.: Ilvr: Conditioning method for denoising diffusion probabilistic models. arXiv preprint arXiv:2108.02938 (2021)
5. Geng, Z., Yang, B., Hang, T., Li, C., Gu, S., Zhang, T., Bao, J., Zhang, Z., Hu, H., Chen, D., et al.: Instructdiffusion: A generalist modeling interface for vision tasks. arXiv preprint arXiv:2309.03895 (2023)
6. Han, L., Wen, S., Chen, Q., Zhang, Z., Song, K., Ren, M., Gao, R., Stathopoulos, A., He, X., Chen, Y., et al.: Proxedit: Improving tuning-free real image editing with proximal guidance. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4291–4301 (2024)
7. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2328–2337 (2023)
8. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
9. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
10. Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: International Conference on Machine Learning. pp. 13213–13232. PMLR (2023)
11. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. arXiv preprint arXiv:2310.01506 (2023)
12. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2426–2435 (2022)
13. Li, W., Yu, X., Zhou, K., Song, Y., Lin, Z., Jia, J.: Sdm: Spatial diffusion model for large hole image inpainting. arXiv preprint arXiv:2212.02963 **1** (2022)
14. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022)
15. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)

16. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Dragondiffusion: Enabling drag-style manipulation on diffusion models. arXiv preprint arXiv:2307.02421 (2023)
17. Nam, H., Kwon, G., Park, G.Y., Ye, J.C.: Contrastive denoising score for text-guided latent diffusion image editing. arXiv preprint arXiv:2311.18608 (2023)
18. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
19. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
20. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
21. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2020)
22. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1921–1930 (2023)
23. Wallace, B., Gokul, A., Naik, N.: Edict: Exact diffusion inversion via coupled transformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22532–22541 (2023)
24. Wang, Q., Zhang, B., Birsak, M., Wonka, P.: Mdp: A generalized framework for text-guided image editing by manipulating the diffusion path. arXiv preprint arXiv:2303.16765 (2023)
25. Xie, S., Zhang, Z., Lin, Z., Hinz, T., Zhang, K.: Smartbrush: Text and shape guided object inpainting with diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22428–22437 (2023)
26. Zhao, J., Zheng, H., Wang, C., Lan, L., Huang, W., Yang, W.: Null-text guidance in diffusion models is secretly a cartoon-style creator. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 5143–5152 (2023)