An Edit Friendly DDPM Noise Space:
Inversion and Manipulations

Inbar Huberman-Spiegelglas   Vladimir Kulikov   Tomer Michaeli

Technion – Israel Institute of Technology

inbarhub@gmail.com, {vladimir.k@campus, tomer.m@ee}.technion.ac.il
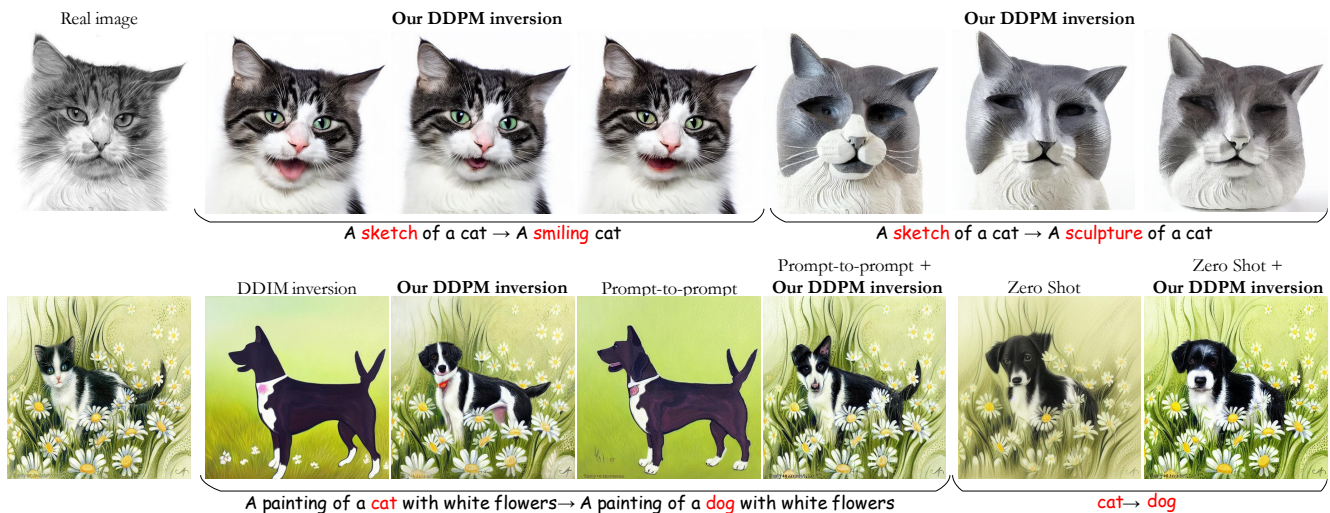
Figure 1. **Edit friendly DDPM inversion.** We present a method for extracting a sequence of DDPM noise maps that perfectly reconstruct a given image. These noise maps are distributed differently from those used in regular sampling, and are more edit-friendly. Our method allows diverse editing of real images without fine-tuning the model or modifying its attention maps, and it can also be easily integrated into other algorithms (illustrated here with Prompt-to-Prompt [9] and Zero-Shot I2I [21])

Abstract

Denoising diffusion probabilistic models (DDPMs) employ a sequence of white Gaussian noise samples to generate an image. In analogy with GANs, those noise maps could be considered as the latent code associated with the generated image. However, this native noise space does not possess a convenient structure, and is thus challenging to work with in editing tasks. Here, we propose an alternative latent noise space for DDPM that enables a wide range of editing operations via simple means, and present an inversion method for extracting these edit-friendly noise maps for any given image (real or synthetically generated). As opposed to the native DDPM noise space, the edit-friendly noise maps do not have a standard normal distribution and are not statistically independent across timesteps. However, they allow perfect reconstruction of any desired image, and simple transformations on them translate into meaningful manipulations of the output image (e.g. shifting, color edits). Moreover, in text-conditional models, fixing those noise maps while changing the text prompt, modifies semantics while retaining structure. We illustrate how this property enables text-based editing of real images via the diverse DDPM sampling scheme (in contrast to the popular non-diverse DDIM inversion). We also show how it can be used within existing diffusion-based editing methods to improve their quality and diversity. The code of the method is attached to this submission.

1. Introduction

Diffusion models have emerged as a powerful generative framework, achieving state-of-the-art quality on image synthesis [2, 11, 20, 22, 23, 25]. Recent works harness diffusion models for various image editing and manipulation tasks, including text-guided editing [1, 5, 9, 14, 27], inpainting [16], and image-to-image translation [17, 24, 30]. A key challenge in these methods is to leverage them for editing of *real* content (as opposed to model-generated images). This

requires inverting the generation process, namely extracting a sequence of noise vectors that would reconstruct the given image if used to drive the reverse diffusion process.

Despite significant advancements in diffusion-based editing, inversion is still considered a major challenge, particularly in the denoising diffusion probabilistic model (DDPM) sampling scheme [11]. Many recent methods (*e.g.* [5, 9, 19, 21, 27, 29]) rely on an approximate inversion method for the denoising diffusion implicit model (DDIM) scheme [26], which is a deterministic sampling process that maps a single initial noise vector into a generated image. However this DDIM inversion method becomes accurate only when using a large number of diffusion timesteps (*e.g.* 1000), and even in this regime it often leads to sub-optimal results in text-guided editing [9, 19]. To battle this effect, some methods fine-tune the diffusion model based on the given image and text prompt [13, 14, 28, 32]. Other methods intervene in the generative process in various ways, *e.g.* by injecting the attention maps derived from the DDIM inversion process into the text-guided generative process [4, 9, 21, 27].

Here we address the problem of inverting the DDPM scheme. As opposed to DDIM, in DDPM, $T+1$ noise maps are involved in the generation process, each of which has the same dimension as the generated output. Therefore, the total dimension of the noise space is larger than that of the output and there exist infinitely many noise sequences that perfectly reconstruct the image. While this property may provide flexibility in the inversion process, not every consistent inversion (*i.e.* one that leads to perfect reconstruction) is also edit friendly. For example, one property we want from an inversion in the context of text-conditional models, is that fixing the noise maps and changing the text-prompt would lead to an artifact-free image, where the semantics correspond to the new text but the structure remains similar to that of the input image. What consistent inversions satisfy this property? A tempting answer is that the noise maps should be statistically independent and have a standard normal distribution, like in regular sampling. Such an approach was pursued in [30]. However, as we illustrate in Fig. 2, this native DDPM noise space is in fact not edit friendly.

Here we present an alternative inversion method, which constitutes a better fit for editing applications, from text-guidance manipulations, to editing via hand-drawn colored strokes. Our inversion "imprints" the image more strongly onto the noise maps, which leads to better preservation of structure when fixing them and changing the condition of the model. This is achieved by the fact that our noise maps have higher variances than the native ones. Our inversion requires no optimization and is extremely fast. Yet, it allows achieving state-of-the-art results on text-guided editing tasks with a relatively small number of diffusion steps, simply by fixing the noise maps and changing the text condition (*i.e.* without requiring model fine-tuning or intervention in the attention

Figure 2. **The native and edit friendly noise spaces.** When sampling an image using DDPM (left), there is access to the "ground truth" noise maps that generated it. This native noise space, however, is not edit friendly (2nd column). For example, fixing those noise maps and changing the text prompt, changes the image structure (top). Similarly, flipping (middle) or shifting (bottom) the noise maps completely modifies the image. By contrast, our edit friendly noise maps enable editing while preserving structure (right).

maps). Importantly, our DDPM inversion can also be readily integrated with existing diffusion based editing methods that currently rely on approximate DDIM inversion. As we illustrate in Fig. 1, this improves their ability to preserve fidelity to the original image. Furthermore, since we find the noise vectors in a stochastic manner, we can provide a diverse set of edited images that all conform to the text prompt, a property not naturally available with DDIM inversion, see top row of Fig. 1 and the Supplementary Supplementary (SM).

2. Related work

2.1. Inversion of diffusion models

Editing a real image using diffusion models requires extracting the noise vectors that would generate that image when used within the generative process. The vast majority of diffusion-based editing works use the DDIM scheme, which is a deterministic mapping from a single noise map to a generated image [5, 7, 9, 19, 21, 27, 29]. The original DDIM paper [26] suggested an efficient approximate inversion for that scheme. This method incurs a small error at every diffusion timestep, and these errors often accumulate into meaningful deviations when using classifier-free guidance [10]. Mokady *et al.* [19] improve the reconstruction quality by fixing each timestep drifting. Their two-step approach first uses DDIM

inversion to compute a sequence of noise vectors, and then uses this sequence to optimize the input null-text embedding at every timestep. Miyake *et al*. [18] achieve similar reconstruction accuracy through forward propagation without optimization, thereby enabling much faster editing processes. An improvement in the reconstruction quality was suggested by Han *et al*. [8] that integrate a regularization term into the null-text embedding optimization. EDICT [29] enables mathematically exact DDIM-inversion of real images by maintaining two coupled noise vectors which are used to invert each other in an alternating fashion. This method doubles the computation time of the diffusion process. CycleDiffusion [30] presents a DDPM-inversion method by recovering a sequence of noise vectors that perfectly reconstruct the image within the DDPM sampling process. As opposed to our method, their extracted noise maps are distributed like the native noise space of DDPM, which results in limited editing capabilities (see Figs. 2,4).

2.2. Image editing using diffusion models

The DDPM sampling method is not popular for editing of real images. When used, it is typically done without exact inversion. Two examples are Ho *et al*. [11], who interpolate between real images, and Meng *et al*. [17] who edit real images via user sketches or strokes (SDEdit). Both construct a noisy version of the real image and apply a backward diffusion after editing. They suffer from an inherent tradeoff between the realism of the generated image and its faithfulness to the original contents. DiffuseIT [15] performs image translation guided by a reference image or by text, also without explicit inversion. They guide the generation process by losses that measure similarity to the original image [6].

A series of papers apply text-driven image-to-image translation using DDIM inversion. Narek *et al*. [27] and Cao *et al*. [4] achieve this by manipulating spatial features and their self-attention inside the model during the diffusion process. Hertz *et al*. [9] change the attention maps of the original image according to the target text prompt and inject them into the diffusion process. DiffEdit [5] automatically generates a mask for the regions of the image that need to be edited, based on source and target text prompts. This is used to enforce the faithfulness of the unedited regions to the original image, in order to battle the poor reconstruction quality obtained from the inversion. This method fails to predict accurate masks for complex prompts.

Some methods utilize model optimization based on the target text prompt. DiffusionCLIP [14] uses model fine-tuning based on a CLIP loss with a target text. Imagic [13] first optimizes the target text embedding, and then optimizes the model to reconstruct the image with the optimized text embedding. UniTune [28] also uses fine-tuning and shows great success in making global stylistic changes and complex local edits while maintaining image structure. Other works
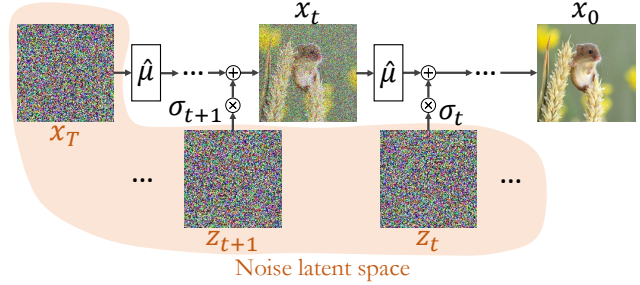

Figure 3. **The DDPM latent noise space.** In DDPM, the generative (reverse) diffusion process synthesizes an image x_0 in T steps, by utilizing $T+1$ noise maps, $\{x_T, z_T, \ldots, z_1\}$. We regard those noise maps as the latent code associated with the generated image.

like Palette [24] and InstructPix2Pix [3], learn conditional diffusion models tailored for specific editing tasks.

3. The DDPM noise space

Here we focus on the DDPM sampling scheme, which is applicable in both pixel space [11] and latent space [23]. DDPM draws samples by attempting to reverse a diffusion process that gradually turns a clean image x_0 into white Gaussian noise,

$$x_t = \sqrt{1-\beta_t}\,x_{t-1} + \sqrt{\beta_t}\,n_t, \quad t=1,\ldots,T, \quad (1)$$

where $\{n_t\}$ are iid standard normal vectors and $\{\beta_t\}$ is some variance schedule. The forward process (1) can be equivalently expressed as

$$x_t = \sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon_t, \quad (2)$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$. It is important to note that in the representation (2), the vectors $\{\epsilon_t\}$ are *not independent*. This is because each ϵ_t corresponds to the accumulation of the noises $n_1, \ldots, n_t$, so that ϵ_t and ϵ_{t-1} are highly correlated for all t. This fact is irrelevant for the training process, which is not affected by the joint distribution of ϵ_t's across different timesteps, but it is important for our discussion below.

The generative (reverse) diffusion process starts from a random noise vector $x_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoises it using the recursion

$$x_{t-1} = \hat{\mu}_t(x_t) + \sigma_t z_t, \quad t=T,\ldots,1, \quad (3)$$

where $\{z_t\}$ are iid standard normal vectors, and

$$\hat{\mu}_t(x_t) = \sqrt{\bar{\alpha}_{t-1}} P(f_t(x_t)) + D(f_t(x_t)). \quad (4)$$

Here, f_t is a neural network that is trained to predict ϵ_t from x_t, $P(f_t(x_t)) = (x_t - \sqrt{1-\bar{\alpha}_t} f_t(x_t))/\sqrt{\bar{\alpha}_t}$ is the predicted x_0, and $D(f_t(x_t)) = \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} f_t(x_t)$ is a

direction pointing to x_t. The variance schedule is taken to be $\sigma_t = \eta\beta_t(1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t)$, where $\eta \in [0, 1]$. The case $\eta = 1$ corresponds to the original DDPM work, and $\eta = 0$ corresponds to the deterministic DDIM scheme.

This generative process can be conditioned on text [23] or class [10] by using a neural network f that has been trained conditioned on those inputs. Alternatively, conditioning can be achieved through guided diffusion [1, 6], which requires utilizing a pre-trained classifier or CLIP model during the generative process.

The vectors $\{x_T, z_T, \ldots, z_1\}$ uniquely determine the image x_0 generated by the process (3) (but not vice versa). We therefore regard them as the latent code of the model (see Fig. 3). Here, we are interested in the inverse direction. Namely, given a real image x_0, we would like to extract noise vectors that, if used in (3), would generate x_0. We refer to such noise vectors as consistent with x_0. Our method, explained next, works with any $\eta \in (0, 1]$.

3.1. Edit friendly inversion

It is instructive to note that *any* sequence of $T + 1$ images $x_0, \ldots, x_T$, in which x_0 is the real image, can be used to extract consistent noise maps by isolating z_t from (3) as[1]

$$z_t = \frac{x_{t-1} - \hat{\mu}_t(x_t)}{\sigma_t}, \quad t = T, \ldots, 1. \quad (5)$$

However, unless such an auxiliary sequence of images is carefully constructed, they are likely to be far from the distribution of inputs on which the network $f_t(\cdot)$ was trained. In that case, fixing the so-extracted noise maps, $\{x_T, z_T, \ldots, z_1\}$, and changing the text condition, may lead to poor results.

What is a good way of constructing auxiliary images $x_1, \ldots, x_T$ for (5) then? A naive approach is to draw them from a distribution that is similar to that underlying the generative process. Such an approach was pursued by [30]. Specifically, they start by sampling $x_T \sim \mathcal{N}(0, \mathbf{I})$. Then, for each $t = T, \ldots, 1$ they isolate ϵ_t from (2) using x_t and the real x_0, substitute this ϵ_t for $f_t(x_t)$ in (4) to compute $\hat{\mu}_t(x_t)$, and use this $\hat{\mu}_t(x_t)$ in (3) to obtain x_{t-1}.

The noise maps extracted by this method are distributed similarly to those of the generative process. Unfortunately, they are not well suited for editing global structures. This is illustrated in Fig. 4 in the context of text guidance and in Fig. 7 in the context of shifts. The reason for this is that DDPM's native noise space is not edit-friendly in the first place. Namely, even if we take a model-generated image, for which we have the "ground-truth" noise maps, fixing them while changing the text prompt does not preserve the structure of the image (see Fig. 2).

Interestingly, here we observe that constructing the auxiliary sequence $x_1, \ldots, x_T$ directly from x_0, and not via (1),

---

[1]Commonly, $z_1 = 0$ in DDPM, so that we run only over $t = T, \ldots, 2$.

Input   Cycle Diffusion   Our inversion

a silhouette of a bird on a branch → a photo of a sparrow on a branch

a photo of a black car → a photo of a red car

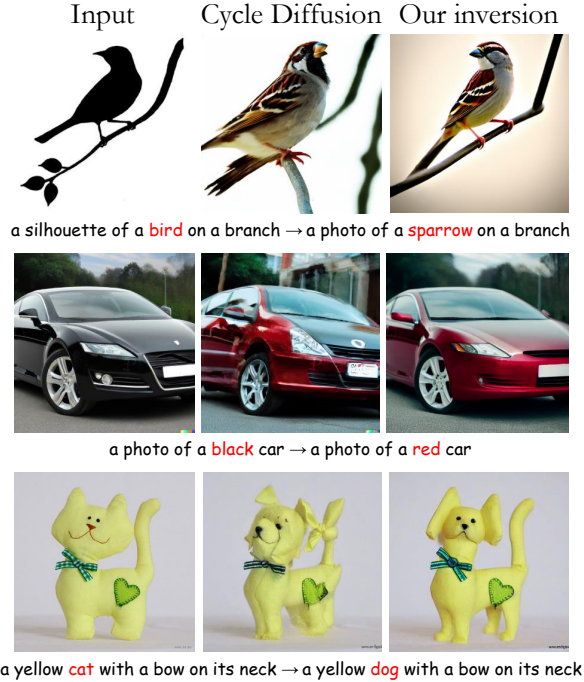a yellow cat with a bow on its neck → a yellow dog with a bow on its neck

Figure 4. **DDPM inversion via CycleDiffusion vs. our method.** CycleDiffusion's inversion [30] extracts a sequence of noise maps $\{x_T, z_T, \ldots, z_1\}$ whose joint distribution is close to that used in regular sampling. However, fixing this latent code and replacing the text prompt fails to preserve the image structure. Our inversion deviates from the regular sampling distribution, but better encodes the image structure.

causes the image x_0 to be more strongly "imprinted" into the noise maps extracted from (5). Specifically, we propose to construct them as

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\,\tilde{\epsilon}_t, \quad 1, \ldots, T, \quad (6)$$

where $\tilde{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ are *statistically independent*. Note that despite the superficial resemblance between (6) and (2), these equations describe fundamentally different stochastic processes. In (2) every pair of consecutive ϵ_t's are highly correlated, while in (6) the $\tilde{\epsilon}_t$'s are independent. This implies that in our construction, x_t and x_{t-1} are typically farther away from each other than in (2), so that every z_t extracted from (5) has a higher variance than in the regular generative process. A pseudo-code of our method is provided in Alg. 1.

A few comments are in place regarding this inversion method. First, it reconstructs the input image up to machine precision, given that we compensate for accumulation of numerical errors (last row in Alg. 1), as we explain in the SM. Second, it is straightforward to use with any kind of diffusion process (*e.g.* a conditional model [11, 12, 23], guided-diffusion [6], classifier-free [10]) by using the appropriate form for $\hat{\mu}_t(\cdot)$. Lastly, due to the randomness in (6), we can obtain many different inversions. While each of

Algorithm 1 Edit-friendly DDPM inversion

Input: real image x_0
Output: $\{x_T, z_T, \ldots, z_1\}$
for $t = 1$ **to** T **do**
   $\tilde{\epsilon} \sim \mathcal{N}(0, 1)$
   $x_t \leftarrow \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \tilde{\epsilon}$
end for
for $t = T$ **to** 1 **do**
   $z_t \leftarrow (x_{t-1} - \hat{\mu}_t(x_t))/\sigma_t$
   $x_{t-1} \leftarrow \hat{\mu}_t(x_t) + \sigma_t z_t$ // to avoid error accumulation
end for
Return: $\{x_T, z_T, \ldots, z_1\}$

them leads to perfect reconstruction, when used for editing they will lead to different variants of the edited image. This allows generating diversity in *e.g.* text-based editing tasks, a feature not naturally available with DDIM inversion methods (see Fig. 1 and SM).

3.2. Properties of the edit-friendly noise space

We now explore the properties of our edit-friendly noise space and compare it to the native DDPM noise space. We start with a 2D illustration, depicted in Fig. 5. Here, we use a diffusion model designed to sample from $\mathcal{N}(\begin{smallmatrix} 10 \\ 10 \end{smallmatrix}), \mathbf{I})$. The top-left pane shows a regular DDPM process with 40 inference steps. It starts from $x_T \sim \mathcal{N}(\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}), \mathbf{I})$ (black dot at the bottom left), and generates a sequence $\{x_t\}$ (green dots) that ends in x_0 (black dot at the top right). Each step is broken down to the deterministic drift $\hat{\mu}_t(x_t)$ (blue arrow) and the noise vector z_t (red arrow). On the top-right pane, we show a similar visualization, but for our latent space. Specifically, here we compute the sequences $\{x_t\}$ and $\{z_t\}$ using Alg. 1 for some given $x_0 \sim \mathcal{N}(\begin{smallmatrix} 10 \\ 10 \end{smallmatrix}), \mathbf{I})$. As can be seen, in our case, the noise perturbations $\{z_t\}$ are larger. This property comes from our construction of x_t, which are typically farther away from one another than in (2). How can the red arrows be longer and still form a trajectory from the origin to the blue cloud? Close inspection reveals that the angles between consecutive noise vectors tend to be obtuse. In other words, our noise vectors are (negatively) correlated across consecutive times. This can also be seen from the two plots in the bottom row, which depict the histograms of angles between consecutive noise vectors for the regular sampling process and for ours. In the former case, the angle distribution is uniform, and in the latter it has a peak at $180°$.

The same qualitative behavior occurs in diffusion models for image generation. Figure 6 shows the per-pixel variance of z_t and correlation between z_t and z_{t-1} for sampling with 100 steps from an unconditional diffusion model trained on Imagenet. Here the statistics were calculated over 10 images drawn from the model. As in the 2D case, our noise vectors have higher variances and they exhibit negative correlations
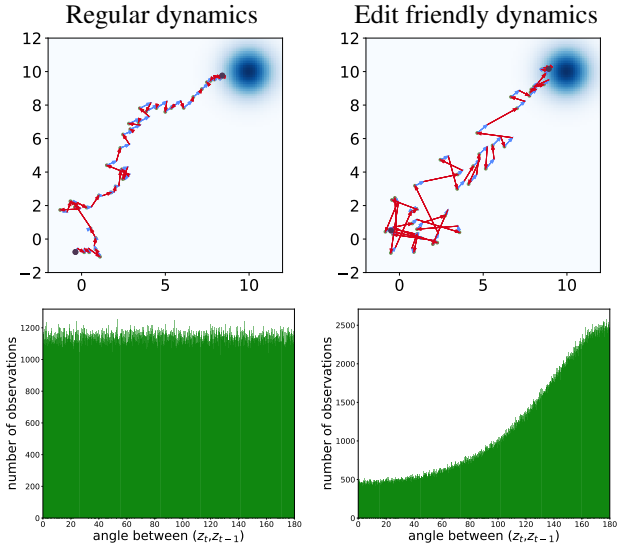

Figure 5. **Regular vs. edit-friendly diffusion.** In the regular generative process (top left), the noise vectors (red) are statistically independent across timesteps and thus the angle between consecutive vectors is uniformly distributed in $[0, 180°]$ (bottom left). In our dynamics (top right) the noise vectors have higher variances and are negatively correlated across consecutive times (bottom right).
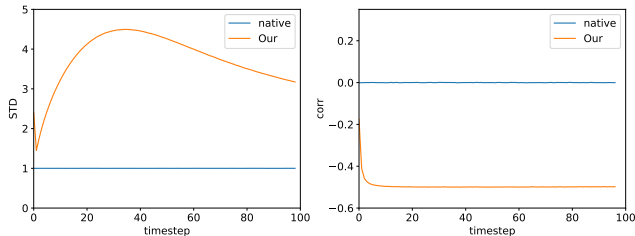

Figure 6. **Native vs. edit friendly noise statistics.** Here we show the per-pixel standard deviations of $\{z_t\}$ and the per-pixel correlation between them for model-generated images.

between consecutive steps. As we illustrate next, these larger variance noise vectors encode the structure of the input image more strongly, and are thus more suitable for editing.

Image shifting Intuitively, shifting an image should be possible by shifting all $T + 1$ maps of the latent code. Figure 7 shows the result of shifting the latent code of a model-generated image by various amounts. As can be seen, shifting the native latent code (the one used to generate the image) leads to a complete loss of image structure. In contrast, shifting our edit-friendly code, results in minor degradation. Quantitative evaluation is provided in the SM.

Color manipulations Our latent space also enables convenient manipulation of color. Specifically, suppose we are given an input image x_0, a binary mask B, and a corresponding colored mask M. We start by constructing $\{x_1, \ldots, x_T\}$
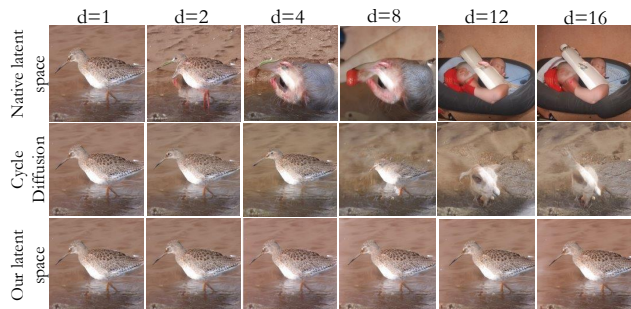
Figure 7. **Image shifting.** We shift to the right a 256×256 image generated by an unconditional model trained on ImageNet by $d = 1 \dots 16$ pixels. When shifting the native noise maps (top) or the ones extracted by CycleDiffusion [30] (middle) the structure is lost. With our latent space, the structure is preserved.

and extracting $\{z_1, \dots, z_T\}$ using (6) and (5), as before. Then, we modify the noise maps as

$$z_t^{\text{edited}} = z_t + sB \odot (M - P(f_t(x_t))), \qquad (7)$$

with $P(f_t(x_t))$ from (4), where s is a parameter controlling the editing strength. We perform this modification over a range of timesteps $[T_1, T_2]$. Note that the term in parenthesis encodes the difference between the desired colors and the predicted clean image in each timestep. Figure 8 illustrates the effect of this process in comparison to SDEdit, which suffers from an inherent tradeoff between fidelity to the input image and conformation to the desired edit. Our approach can achieve a strong editing effect without modifying textures (neither inside nor outside the mask).

4. Text-Guided Image Editing

Our latent space can be utilized for text-driven image editing. Suppose we are given a real image x_0, a text prompt describing it p_{src}, and a target text prompt p_{tar}. To modify the image according to these prompts, we extract the edit-friendly noise maps $\{x_T, z_T, \dots, z_1\}$, while injecting p_{src} to the denoiser. We then fix those noise maps and generate an image while injecting p_{tar} to the denoiser. We run the generation process starting from timestep $T - T_{\text{skip}}$, where T_{skip} is a parameter controlling the adherence to the input image. Figures 1, 2, 4, and 9 show several text driven editing examples using this approach. As can be seen, this method nicely modifies semantics while preserving the structure of the image. In addition, it allows generating diverse outputs for any given edit (see Fig. 1 and SM). We further illustrate the effect of using our inversion in combination with methods that rely on DDIM inversion (Figs. 1 and 11). As can be seen, these methods often do not preserve fine textures, like fur, flowers, or leaves of a tree, and oftentimes also do not preserve the global structure of the objects. By integrating our inversion, structures and textures are better preserved.
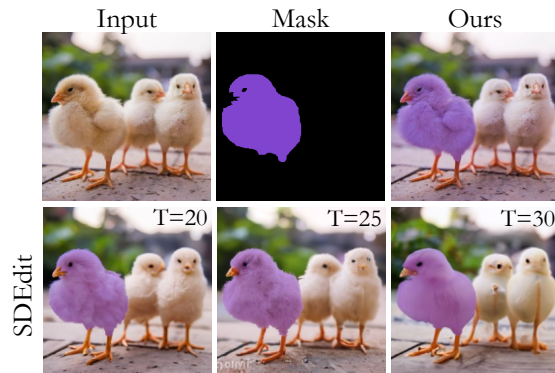

Figure 8. **Color manipulation on a real image.** Our method (applied here from $T_2 = 70$ to $T_1 = 20$ with $s = 0.05$) leads to a strong editing effect without modifying textures and structures. SDEdit, on the other hand, either does not integrate the mask well when using a small noise (left) or does not preserve structures when the noise is large (right). In both methods, we use an unconditional model trained on ImageNet with 100 inference steps.

5. Experiments

We evaluate our method both quantitatively and qualitatively on text-guided editing of real images. We analyze the usage of our extracted latent code by itself (as explained in Sec. 4), and in combination with existing methods that currently use DDIM inversion. In the latter case, we extract the noise maps using our DDPM-inversion and inject them in the reverse process, in addition to any manipulation they perform on *e.g.* attention maps. All experiments use a real input image as well as source and target text prompts.

Implementation details We use Stable Diffusion [23], in which the diffusion process is applied in the latent space of a pre-trained image autoencoder. The image size is $512 \times 512 \times 3$, and the latent space is $64 \times 64 \times 4$. Our method is also applicable in unconditional pixel space models using CLIP guidance, however we found Stable Diffusion to lead to better results. Two hyper-parameters control the balance between faithfulness to the input image and adherence to the target prompt: the strength of the classifier-free guidance [10], and T_{skip} explained in Sec. 4. In all our numerical analyses and all the results in the paper, we used strength $= 15$, $T_{\text{skip}} = 36$, $\eta = 1$ and 100 inference steps, unless noted otherwise. In the SM we provide a thorough analysis of the effect of the strength and T_{skip} hyper-parameters.

Datasets We use two datasets of real images: (i) "modified ImageNet-R-TI2I" from [27] with additional examples collected from the internet and from other datasets, (ii) "modified Zero-Shot I2IT", which contains images of 4 classes (Cat, Dog, Horse, Zebra) from [21] and from the internet. The first dataset comprises 48 images, with 3-5 different target prompts for each, leading to 212 image-text pairs. The

| Input | PnP | EDICT | null-text inv. | CycleDiffusion | Our inv. |
| Time [sec] | [206] | [520] | [160] | [36] | [36] |

A bear doll with a blue knitted sweater → A bear doll with a blue knitted sweater and a hat

An origami of a hummingbird → A sculpture of a hummingbird

A photo of a horse in the mud → A photo of a zebra in the snow

A photo of a cat sitting on a car → A photo of a smiling dog sitting on a car
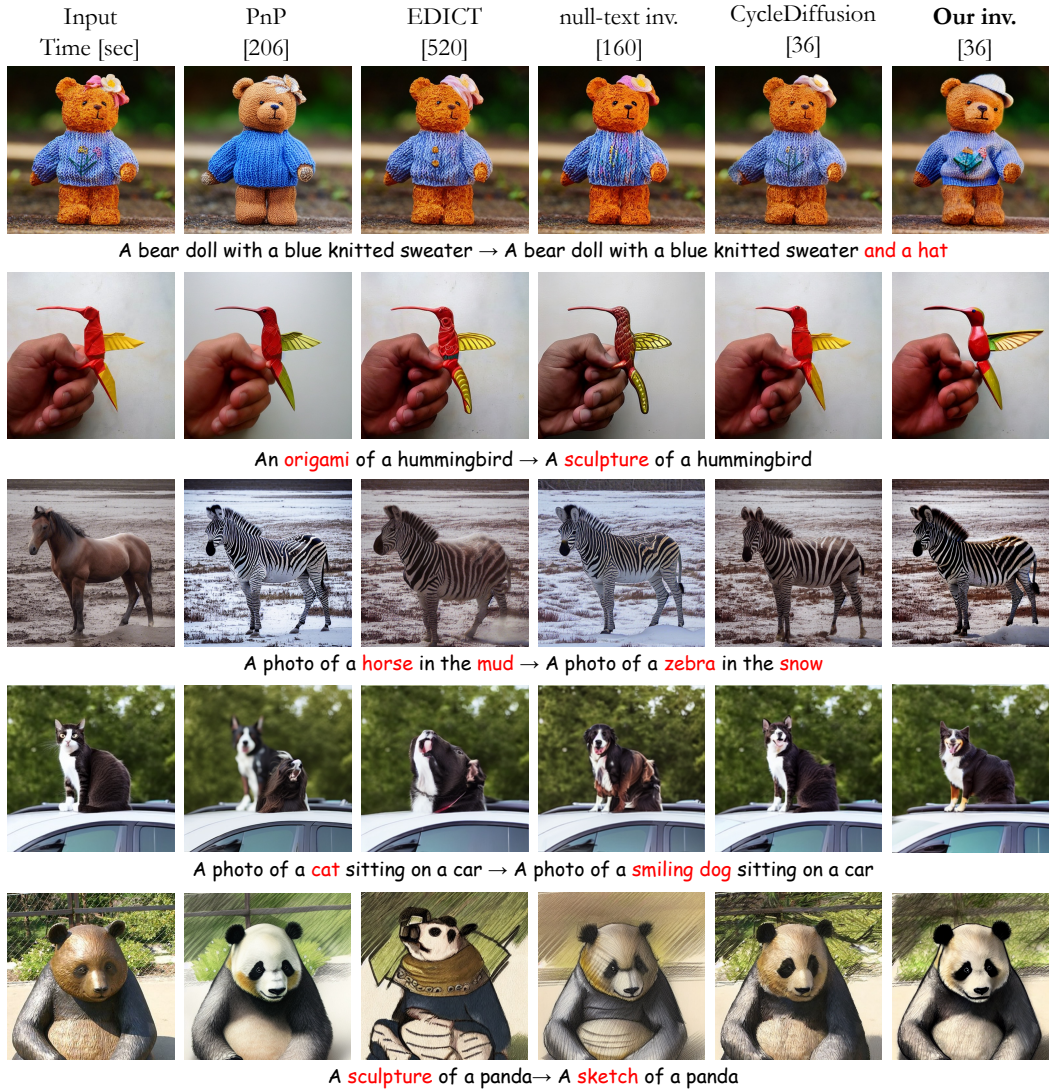
A sculpture of a panda→ A sketch of a panda

Figure 9. **Comparisons.** We show results for editing of real images using all methods. Our approach maintains high fidelity to the input while conforming to the text prompt. The time taken to edit a single image is indicated within parentheses.

second dataset has 15 images in each category with one target prompt for each, making 60 image-text pairs in total. Please refer to the SM for full details.

Metrics We numerically evaluate the results using two complementary metrics: LPIPS [31] to quantify the extent of structure preservation (lower is better) and a CLIP-based score to quantify how well the generated images comply with the text prompt (higher is better). We additionally quantify the editing time in seconds required for image editing. Further information on diversity is provided in the SM.

Comparisons on the modified ImageNet-R-TI2I dataset We perform comparisons with Plug-and-Play (PnP) [27], EDICT [29], null-text inversion [19] and CycleDiffusion [30]. Additionally, we assess our method against

prompt-to-prompt (p2p) [9], both as a standalone technique and when integrated with our inversion. Further details on the integration can be found in the SM. We report the results of CycleDiffusion with $\eta = 1$, similarly to our configuration. Quantitative results with $\eta = 0.1$, as suggested in their paper, can be found in SM. Finally, we compare our method to both plain DDIM inversion and DDIM inversion that applies the inversion until a specific timestep. All methods were run with the default parameters suggested by their authors and are provided in SM. For a fair comparison, we use the same parameters of CycleDiffusion across all images within the experiment. This is in contrast to their paper where parameters are individually selected for each image.

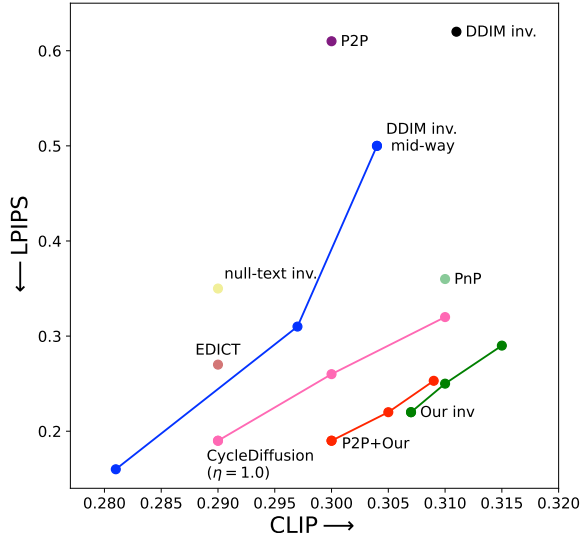As seen in Fig. 9, our method successfully modifies real

Figure 10. **Fidelity to source image vs. compliance with target text.** The plot compares the LPIPS and CLIP scores achieved by all methods on the modified ImageNet-R-TI2I dataset. Our inversion, P2P with our inversion, and CycleDiffusion are shown with three options of the parameters (strength,T_{skip}): for our method $(15, 36), (12, 36), (9, 36)$, for P2P+Ours $(7.5, 8), (7.5, 12), (9, 20)$, and for CycleDiffusion $(3, 30), (4, 25), (4, 15)$. DDIM inversion mid-way is shown with three options for T_{skip}: $20, 40, 60$, all with a guidance strength of 9. The parameters for the other methods are reported in the SM. For CLIP higher is better, while for LPIPS lower is better.

images according to the target prompts. In all cases, our results exhibit both high fidelity to the input image and adherence to the target prompt. EDICT shows some artifacts in their results and CycleDiffusion produces images with less compliance with the target text. PnP and null-text inversion often preserve structure but require more than 2.5 minutes to edit an image. Qualitative results with plain DDIM inversion and P2P with and without our inversion appear in SM. Figure 10 shows the CLIP-LPIPS losses graph for all methods, where, for our inversion, P2P with our inversion, and CycleDiffusion we report these losses with three different parameters. As can be seen, our method achieves a good balance between LPIPS and CLIP. CycleDiffusion struggles to apply a strong edit while preserving the structure. Integrating our inversion into P2P improves their performance in both metrics. See more details in the SM.

Comparisons on the modified Zero-Shot I2IT dataset
Next, we compare our method to Zero-Shot Image-to-Image Translation (Zero-Shot) [21], which uses DDIM inversion. This method only translates between several predefined classes. We follow their setting and use 50 diffusion steps. When using with our inversion, we decrease the hyperparameter controlling the cross-attention from the default
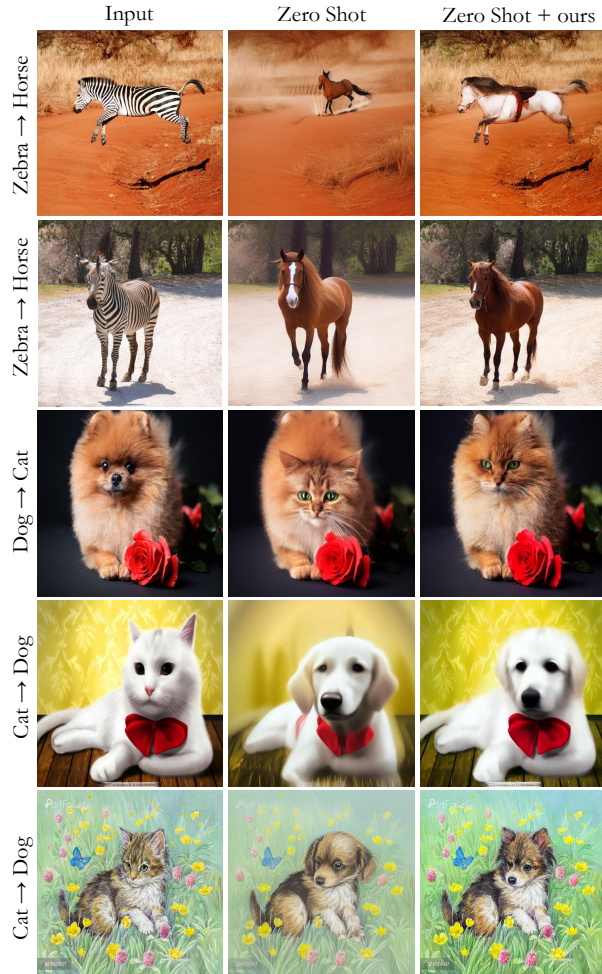

Figure 11. **Improving Zero-Shot I2I Translation.** Images generated by Zero Shot I2I suffer from loss of detail. With our inversion, fine textures, like fur and flowers, are retained. Both methods achieve CLIP accuracy of 0.88, however, Zero-Shot method achieves LPIPS score of 0.35 while Zero-Shot with our inversion produces images more similar to the input, and hence, achievs an LPIPS score of 0.27 (for LPIPS lower is better).

value 0.1 to 0.03. As can be seen in Fig. 11, while Zero-Shot's results comply with the target text, they are typically blurry and miss detail. Integrating our inversion adds back the details from the input image. See more details in the SM.

6. Conclusion

We presented an inversion method for DDPM. Our noise maps encode the image structure more strongly than the noise maps in regular sampling, and are therefore better suited for image editing. We illustrated their advantages in text-based editing, both when used by themselves and in combination with other editing methods.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, 2022. 1, 4

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 1

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, 2023. 3

[4] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 22560–22570, 2023. 2, 3

[5] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 3

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:8780–8794, 2021. 3, 4

[7] Adham Elarabawy, Harish Kamath, and Samuel Denton. Direct inversion: Optimization-free text-driven real image editing with diffusion models. *arXiv preprint arXiv:2211.07825*, 2022. 2

[8] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Yuxiao Chen, Di Liu, Qilong Zhangli, et al. Improving negative-prompt inversion via proximal guidance. *arXiv preprint arXiv:2306.05414*, 2023. 3

[9] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 3, 7

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 4, 6

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 1, 2, 3, 4

[12] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23:47:1–47:33, 2021. 4

[13] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[14] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2416–2425, 2022. 1, 2, 3

[15] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation. *arXiv preprint arXiv:2209.15264*, 2022. 3

[16] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461, 2022. 1

[17] 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 (ICLR)*, 2022. 1, 3

[18] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 3

[19] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6038–6047, 2023. 2, 7

[20] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning (ICLR)*, pages 16784–16804. PMLR, 2022. 1

[21] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 1, 2, 6, 8

[22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1, 3, 4, 6

[24] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. *ACM SIGGRAPH 2022 Conference Proceedings*, 2021. 1, 3

[25] 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 (NeurIPS)*, 35:36479–36494, 2022. 1

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising

diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[27] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2023. 1, 2, 3, 6, 7

[28] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. UniTune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022. 2, 3

[29] Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22532–22541, 2023. 2, 3, 7

[30] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 4, 6, 7

[31] 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 (CVPR)*, pages 586–595, 2018. 7

[32] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6027–6037, 2023. 2