

LUSD: Localized Update Score Distillation for Text-Guided Image Editing

Worameth Chinchuthakun^{*†1,2}
Pitchaporn Rewatbowornwong¹

Tossaporn Saengja^{*†1,3}
Pramook Khungnurn⁴

Nontawat Tritrong¹
Supasorn Suwajanakorn¹

¹ VISTEC

² Siam Commercial Bank

³ Faculty of Medicine Siriraj Hospital

⁴ Pixiv

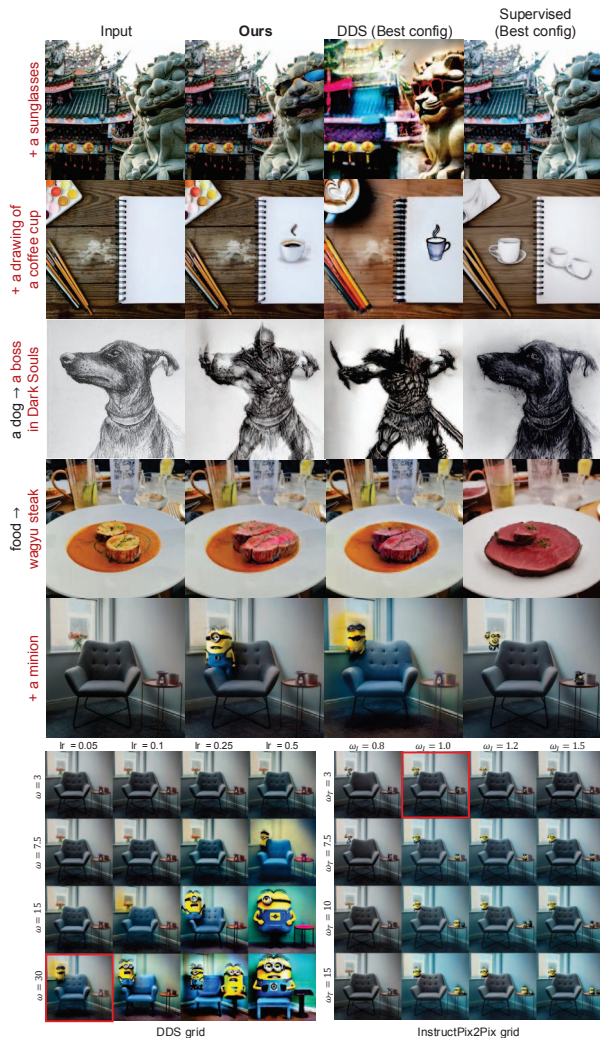


Figure 1. We propose a novel score distillation technique for object insertion and image editing tasks. Compared to existing score distillation methods, (e.g., DDS [16]) and supervised methods (e.g., InstructPix2Pix [6]), our LUSD achieves a higher success rate with superior background preservation. We compare the best results (prioritizing object appearance) from state-of-the-art methods, optimized through hyperparameter tuning (highlighted by red boxes), with our results all generated using a *single* configuration. Hypertuning grids for other images are in Appendix 12.

Abstract

While diffusion models show promising results in image editing given a target prompt, achieving both prompt fidelity and background preservation remains difficult. Recent works have introduced score distillation techniques that leverage the rich generative prior of text-to-image diffusion models to solve this task without additional fine-tuning. However, these methods often struggle with tasks such as object insertion. Our investigation of these failures reveals significant variations in gradient magnitude and spatial distribution, making hyperparameter tuning highly input-specific or unsuccessful. To address this, we propose two simple yet effective modifications: attention-based spatial regularization and gradient filtering-normalization, both aimed at reducing these variations during gradient updates. Experimental results show our method outperforms state-of-the-art score distillation techniques in prompt fidelity, improving successful edits while preserving the background. Users also preferred our method over state-of-the-art techniques across three metrics, and by 58-64% overall.

1. Introduction

In the problem of *text-guided image editing*, we are given an image and a text prompt, and the goal is to modify the image to match the prompt. Unlike image generation, editing requires preserving elements of the input image, such as a fox’s face when *adding sunglasses*, a cat’s outline when *changing its color*, and the overall structure of an outdoor scene when *transitioning it from summer to winter*.

Recent approaches leverage large-scale text-to-image diffusion models [18, 40], such as Stable Diffusion [37], to tackle the task. Supervised methods [7, 39, 51] fine-tune or train diffusion models on large-scale synthetic datasets conditioned on edit instructions, enabling intuitive user interaction via natural language prompts. For example, a user may provide an image and ask the system to “*add a cat on the sofa*”. In contrast, zero-shot methods [5, 13, 15, 27, 34] at-

^{*} Authors contributed equally to this work.

[†] Work done during research assistantships at VISTEC.

tempt to invert the diffusion process of the input image and *regenerate* it with a new prompt. Recently, Score Distillation Sampling (SDS) [36] has emerged as a promising alternative. Instead of relying on additional training data or diffusion inversion, SDS leverages the prior from pre-trained diffusion models to optimize the input image to align with the text prompt. DDS [16] extends SDS by reducing noisy gradient directions, which helps preserve the original content and can be enhanced with additional regularization to better maintain structural information [30].

Despite many attempts, editing an image to match a prompt while preserving the background remains challenging. Supervised methods typically perform well only in limited scenarios, as they rely on small or synthetic training sets, which can be biased and fail to capture the diversity of real-world cases (Figure 1). Notably, these methods also struggle to preserve the background—a limitation shared by most zero-shot methods as they often depend on inferred implicit binary masks that can be inaccurate. While SDS-based methods can preserve the background better, they sometimes fail to insert objects altogether (see Figure 2). Moreover, they only work within a narrow range of hyperparameters, requiring tuning for each input image. Object insertion, which involves deciding *where* and *how* to generate objects from scratch, poses even greater challenges for these methods, especially for unusual combinations, such as “a Chinese lion statue wearing sunglasses” (Figure 1).

In this paper, we investigate a solution based on score distillation and its associated challenges. For such a method, an input image is gradually transformed through gradient updates derived from the denoising process of a text-conditioned diffusion model. One difficulty lies in the extreme variations in gradient magnitudes, which make it difficult to determine the correct learning rate or apply a regularizer to preserve the background. These variations can come from simply changing the prompt or the input image. Moreover, even when the text prompt and image are fixed, the denoising process with different noise seeds still strongly influences the gradient magnitude and its spatial distributions [24], leading to gradients in multiple locations counteracting each other’s progress.

Our method, Localized Update Score Distillation (LUSD), builds upon a score-distillation formulation [26] with a simple L_2 regularizer that pulls updates toward the original image and incorporates two key ideas. First, to reduce variation in spatial distributions, we track the spatial locations of the edits made by SDS using attention-based features. By computing a moving average of these estimated locations during optimization and using it to modulate the gradients, updates progressively focus on narrower areas, increasing the rate at which new objects appear and allowing the background to be preserved better. Second, we implement a normalization and thresholding mechanism to

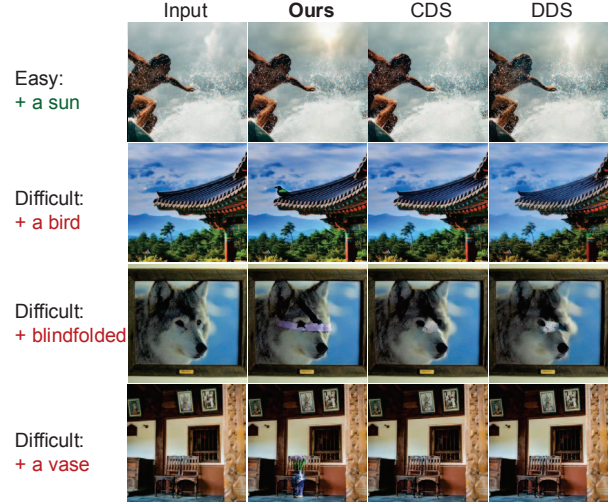


Figure 2. CDS [30] and DDS [16] tend to generate incomplete objects, or completely fail to produce one. Our method mitigates these issues while preserving the background.

filter out “counterproductive” gradients, identified by their low standard deviation.

We evaluate our method against InstructPix2Pix [6], HIVE [51], LEDITS++ [5], DDS [16], and CDS [30] on a standard image editing benchmark, MagicBrush [49]. Our method achieves a higher success rate in object addition and preserves the original background more effectively than the competitors. Additionally, it retains the general editing capabilities inherent to score distillation.

To summarize, our contributions are:

- An analysis of gradient behavior, identifying key factors that hinder effective object insertion in score distillation.
- A novel attention-based spatial regularization and gradient normalization for mitigating *bad gradients* effects.

2. Related Work

Diffusion-based text-guided image editing. While there exist image editing techniques that require binary masks [2, 3, 43, 46, 47, 53] or other conditions [50], we focus on approaches that do not require such explicit spatial conditions, categorized into zero-shot and supervised methods.

Most zero-shot methods first invert the input image along its diffusion trajectory conditioned on a caption and then denoise the result using a new target caption. The inversion process can be achieved with DDIM [40], DDPM [18, 19], optimized text embeddings [28, 29], DPM-Solver++ [5], or simply adding noise as in SDEdit [27]. TiNO-Edit [12] proposes a strategy to determine the optimal timestep and noise required for this process. To better preserve content of the input image, some methods derive an implicit mask to guide the editing process. DiffEdit [13] infers such a mask by computing the noise difference when conditioning the model on the source and target texts. Other works

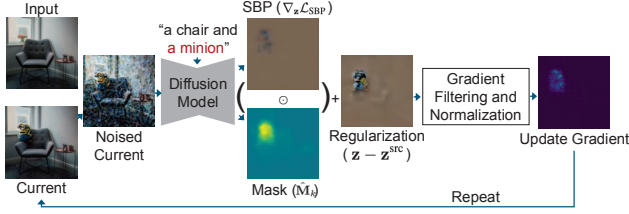


Figure 3. An overview of our method. Given an input image and a target prompt, we obtain gradient of the SBP loss [26] and an attention-based mask. With spatial regularization, gradient filtering and normalization, we modify the image to match the prompt.

[9, 15, 34, 41] extract attention features of the noised image during the inversion process, then inject them to preserve the original structure when denoising. LEDITS++ [5] combines both attention-based and noise-based methods to generate a more precise editing mask. Some works [38, 42] focus on rectified flow models based on the MM-DiT architecture [14]; however, they are incompatible with diffusion models as they depend on rectified ODE properties [38] or MM-DiT’s multi-modal self-attention [42].

Another line of work trains diffusion models conditioned on an edit instruction on large-scale synthetic datasets for general-purpose editing models. InstructPix2Pix [7] synthesizes their training samples using Prompt-to-Prompt [15], whereas Hive [51] enhances this pipeline by fine-tuning with human feedback. Emu Edit [39] improves data generation by localizing the editing area with a more accurate mask derived from large language models. It also introduces task-specific embeddings and trains the model on a wider range of tasks, resulting in better generalization.

Still, editing an image to align with a target text prompt without altering unrelated regions remains challenging. Inversion-based methods contain no explicit constraints to preserve the background, except for an approximated mask, and instruction-based supervised methods suffer from imperfect, synthetic datasets. Our method is based on score distillation, an optimization-based approach that can easily incorporate a loss term to keep the background intact.

Score distillation. Score distillation technique has shown promising performance in text-to-3D generation, although with some flaws. The original formulation, SDS [36], often produces blurry and over-saturated outputs due to its mode-seeking behavior and requires high classifier-free guidance (CFG) values [17]. Subsequent formulations [1, 20, 26, 44, 48] addressed these issues to improve output quality. Recent works [8, 16, 21, 22, 30] have also adapted these techniques to 2D image editing. DDS [16] leverages the source caption and the input image to reduce noisy gradients by using the difference between the gradients of the target pair and the source pair. CDS [30] builds upon DDS by regularizing structural changes with a CUT loss [33] derived from the self-attention features of the diffusion model

to better preserve the source image’s structure. DreamSampler [21] explores data consistency terms controlled by a weighting hyperparameter λ and improves noise schedules through reverse diffusion sampling.

Although DDS and CDS excel at object replacement and global attribute manipulation (e.g., color, style), they struggle with inserting new objects (Figure 2), as background preservation is not part of their optimization objectives. While DreamSampler explores regularizers, conditioning solely on λ can be sensitive to variations in gradient magnitudes. Our method mitigates these variations, enhancing object insertion while preserving the input image.

3. Approach

Given an input *source image* and a *target prompt* describing how the image should be modified, our goal is to modify the image to match the prompt. We focus on modifications that preserve parts of the original image, avoiding a complete transformation. For example, prompts may add an object, such as “a cat *wearing a hat*”, or adjust global features while retaining the image’s structure, such as “a city *in winter*” to add snow and make the sky cloudy.

Our method, LUSD, is based on score distillation sampling [36]. We introduce regularization techniques to better preserve the background and a method to filter and normalize updates so that the optimization process become robust under diverse inputs without needing instance-specific hyperparameters. An overview is shown in Figure 3. We begin with a review of SDS and explain our choice of an SDS-based method we extend (Section 3.1). Next, we introduce the loss term for background preservation (Section 3.2) and then propose an attention-based spatial regularization (Section 3.3) and a gradient filtering-normalization technique (Section 3.4) that improve the algorithm’s reliability.

3.1. Preliminaries

Diffusion models [18] form a family of generative models that learn a target data distribution p_{data} by transforming samples from a noise distribution. A diffusion model ϵ_ϕ is trained to predict noise $\epsilon \sim \mathcal{N}(0, I)$ that is used to generate a noisy sample $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x} + \sqrt{1 - \alpha_t}\epsilon$ where $\mathbf{x} \sim p_{\text{data}}$ and α_t denotes the noise schedule of the model. The training loss is given by

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}, t, \epsilon} [\|\epsilon_\theta(\mathbf{x}_t, y, t) - \epsilon\|_2^2], \quad (1)$$

where y denotes conditioning signals such as text. In this paper, we use Stable Diffusion [37], which operates on noisy latent codes $\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z} + \sqrt{1 - \alpha_t}\epsilon$ where $\mathbf{z}$ is the (noise-free) latent code of a (noise-free) image $\mathbf{x}$. Note that $\mathbf{x}$ and $\mathbf{z}$ are related by $\mathbf{x} = \mathcal{D}(\mathbf{z})$ and $\mathbf{z} = \mathcal{E}(\mathbf{x})$ where $\mathcal{D}$ and $\mathcal{E}$ are the decoder and encoder of a variational autoencoder (VAE), respectively. Our method optimizes $\mathbf{z}$, which finally must be converted to an image with $\mathcal{D}$.

Score distillation sampling [36] uses ϵ_ϕ to optimize a set of parameters θ that gets converted to a (noise-free) latent code $\mathbf{z}$ through a differentiable function $\mathbf{z} = g(\theta)$, given a text condition y . The loss function is defined implicitly by setting its gradient with respect to θ to be the gradient of (1), also with respect to θ , without the Jacobian term $\partial\epsilon_\phi^\omega/\partial\mathbf{z}_t$:

$$\nabla_\theta \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t,\epsilon} \left[\left(\epsilon_\phi^\omega(\mathbf{z}_t, y, t) - \epsilon \right) \frac{\partial \mathbf{z}_t}{\partial \theta} \right], \quad (2)$$

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

where ω is the classifier-free guidance (CFG) [17] scale. DDS [16] extends this formulation to transform the latent code of a source image $\mathbf{z}^{\text{src}} = \mathcal{E}(\mathbf{x}^{\text{src}})$ into a target latent code $\mathbf{z}$, which aligns with a target text prompt y^{tgt} while preserving the source content. DDS simplifies the problem by setting $\theta = \mathbf{z}$ and g to the identity function. The DDS loss replaces the noise ϵ in (2) with the noise predicted from $\mathbf{z}_t^{\text{src}} = \sqrt{\alpha_t}\mathbf{z}^{\text{src}} + \sqrt{1 - \alpha_t}\epsilon$ and the source prompt y^{src} :

$$\nabla_{\mathbf{z}} \mathcal{L}_{\text{DDS}} = \mathbb{E}_{t,\epsilon} \left[\left(\epsilon_\phi^\omega(\mathbf{z}_t, y^{\text{tgt}}, t) - \epsilon_\phi^\omega(\mathbf{z}_t^{\text{src}}, y^{\text{src}}, t) \right) \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}} \right].$$

The source prompt can be either provided by the user or automatically generated by vision-language models such as BLIP [23] or Chat-GPT [32].

While DDS reduces noisy gradient directions, according to McAllister *et al.* [26], it does not achieve an accurate estimation of the source distribution because the ϵ_ϕ^ω term is not computed from $\mathbf{z}$, the latent code being optimized. As such, we adopt their SBP loss, which replaces $\mathbf{z}_t^{\text{src}}$ with $\mathbf{z}_t$:

$$\nabla_{\mathbf{z}} \mathcal{L}_{\text{SBP}} = \mathbb{E}_{t,\epsilon} \left[\left(\epsilon_\phi^\omega(\mathbf{z}_t, y^{\text{tgt}}, t) - \epsilon_\phi^\omega(\mathbf{z}_t, y^{\text{src}}, t) \right) \frac{\partial \mathbf{z}_t}{\partial \mathbf{z}} \right].$$

3.2. Regularization for background preservation

For many image editing tasks, such as inserting objects or transforming certain elements, it is essential to preserve background areas unrelated to the edit. However, previous SDS-based methods often lack explicit background constraints, leading to unintended changes. As previously explored in DDS [16] and DreamSampler [21], a straightforward solution is to add a regularizing term:

$$\nabla_{\mathbf{z}} \mathcal{L}_{\text{SBP-reg}} = (1 - \lambda)\nabla_{\theta} \mathcal{L}_{\text{SBP}} + \lambda(\mathbf{z} - \mathbf{z}^{\text{src}}). \quad (3)$$

An issue with this formulation is that it is sensitive to the hyperparameter λ , whose optimal value can vary between inputs as different images and prompts naturally yield different gradients. An additional source of variability in tasks like object insertion is whether the object to be added has a single primary location, like a hat on a person, or multiple plausible locations, like a hat on a table. In this example, the gradient $\mathcal{L}_{\text{SBP}}$ averaged over multiple optimization steps for a hat on a table will be much weaker than of a hat on the person’s head (Figure 4). Moreover, even with a fixed image and prompt, different sampled noises ϵ yield gradients with widely varying intensities and spatial distributions.

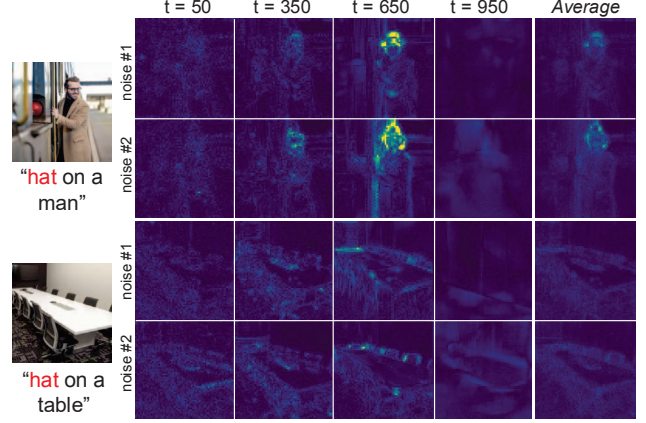


Figure 4. Gradients vary with different timesteps, noises, prompts, images, as well as the number of plausible placements for “hat.”

3.3. Attention-based spatial regularization

Our idea is to use a fixed λ value and to modulate the gradient $\nabla_{\mathbf{z}} \mathcal{L}_{\text{SBP}}$ in (3) so that it becomes low in areas that are likely to be the background. In such an area, the regularizing term will dominate, and the content there is thus encouraged to be the same as that in the source image.

To modulate the gradient, we estimate a mask of the edited region (i.e., the inverse of background region) using a technique called self-attention exponentiation [31], where the mask is computed from attention maps inside the diffusion model’s intermediate activations. Specifically, we assume that the diffusion model ϵ_ϕ is a U-Net with L self-attention layers, and each self-attention layer is immediately followed by a cross-attention layer [37]. When we compute $\epsilon_\phi(\mathbf{z}_t, y^{\text{tgt}}, t)$ in each optimization step, we extract the l^{th} self-attention maps $\mathbf{A}_S^{l,t}$ and a set of associated cross-attention maps $\{\mathbf{A}_C^{l,t,e}\}$ where e denotes a token in the text embedding of the target prompt y^{tgt} . In our case, we extract such maps for all *noun* tokens e associated with the edit, as in [11], which can be inferred by comparing the source and target prompts (see Appendix 6). We then average these maps across all layers to obtain $\mathbf{A}_S^t$ and $\{\mathbf{A}_C^{t,e}\}$. Let N denote the spatial size of the tensor from which the largest self-attention map is computed.¹ We may view $\mathbf{A}_S^t$ as an $N \times N$ matrix, and each $\mathbf{A}_C^{t,e}$ as an $N \times 1$ vector. We then compute the enhanced cross-attention map $\hat{\mathbf{A}}_C^{t,e} = \mathbf{A}_S^t \mathbf{A}_C^{t,e}$ as a matrix-vector product. Finally, we average $\hat{\mathbf{A}}_C^{t,e}$ across all noun tokens e to obtain $\hat{\mathbf{A}}_C^t$. As shown in Figure 5, this method produces cross-attention maps with greater contrast and sharper boundaries, better highlighting edited areas.

In practice, the raw values of $\hat{\mathbf{A}}_C^t$ vary across different image-text pairs. To address this, following [31], we apply min-max normalization to transform each pixel in $\hat{\mathbf{A}}_C^t$ to the range $[0, 1]$, yielding a normalized mask $\mathbf{M}$. During

¹For Stable Diffusion, $N = 1024 = 32 \times 32$.



Figure 5. Refining cross-attention using self-attention [31] produces editing masks with higher contrast and sharper boundaries.

optimization, we compute a moving average of the mask as $\mathbf{M}_k = (1 - \alpha)\mathbf{M}_{k-1} + \alpha\mathbf{M}$, where k is the optimization step and $\mathbf{M}_1 = \mathbf{M}$, to prevent sudden changes in the estimated mask. The gradient update at the k -th step is given by:

$$\begin{aligned}\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP-reg}} &= (1 - \lambda)(\hat{\mathbf{M}}_k \odot \nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP}}) + \lambda(\mathbf{z} - \mathbf{z}^{\text{src}}), \\ \hat{\mathbf{M}}_k &= \beta\mathbf{M}_k + (1 - \beta)\mathbf{1},\end{aligned}$$

where $\odot$ denotes element-wise product, $\mathbf{1}$ is an vector of ones with the same size as $\mathbf{M}$, and β is a hyperparameter controlling the effect of $\mathbf{M}_k$. We found that linearly increasing β from 0 to 1 during optimization generally suffices.

3.4. Gradient filtering and normalization

Another challenge we observe is that some noise samples ϵ produce gradients with very low magnitudes that scatter across the image and fail to drive meaningful progress. When combined with regularization, these weak gradients can even cause the optimized image to revert to the input, as the regularization overpowers $\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP}}$. Such gradients tend to be less localized, and pixel values of $\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP}}$ tend to have a small standard deviation (Figure 6).

To prevent reversion, we detect the above “bad” gradient using a simple test: if the standard deviation of the pixel values of $\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP}}$ is below a certain threshold η , then the gradient is bad. When a bad gradient is found, we repeatedly sample a new noise ϵ while keeping the timestep t constant until a “good” gradient is found for that step of the optimization process. While the range of standard deviations for good gradients can vary across images, our goal here is to avoid problematic ones and allow any sufficiently good gradients to make changes to the image. To ensure that progress is made even when η is set too high, we begin the optimization process with an initial threshold η_0 and exponentially decay it whenever a gradient fails the test. Once a good gradient is found, we reset the threshold back to η_0 .

Finally, to ensure a consistent optimization process and avoid issues with small gradients stalling progress or large gradients causing instability and highly saturated outputs, we normalize the gradients with its standard deviation:

$$\nabla_{\mathbf{z}}\mathcal{L}_{\text{LUSD}} = \gamma \frac{\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP-reg}}}{\text{SD}(\nabla_{\mathbf{z}}\mathcal{L}_{\text{SBP-reg}})}. \quad (4)$$

where γ is a hyperparameter that enables annealing of the optimization process to gradually decrease fluctuation.

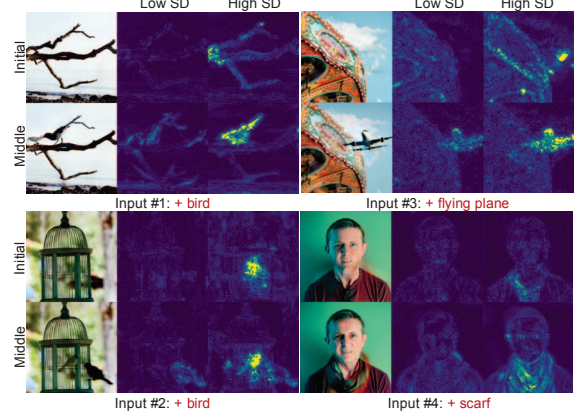


Figure 6. Example gradients with low or high standard deviations sampled from the beginning or middle of the optimization process. Low-SD gradients, which are less focused and counterproductive, are filtered out as they can revert the optimized image to the input.

4. Experiments

Implementation details. We initialize the latent code $\mathbf{z}$ with $\mathbf{z}^{\text{src}} = \mathcal{E}(\mathbf{x}^{\text{src}})$ and use our method to optimize it with the SGD optimizer (PyTorch’s `torch.optim.SGD`) for 300 steps with a learning rate of 2000. Then, we decode $\mathbf{z}$ to get the output $\mathbf{x} = \mathcal{D}(\mathbf{z})$. In each optimization step, we sample timestep $t \sim U(50, 950)$ and set the CFG scale ω to 0. To preserve the background (Section 3.3), we set $\lambda = 0.02$ and compute mask $\mathbf{M}_k$ with $\alpha = 0.1$. Following [31], we extract cross- and self-attention maps from layers with the spatial resolution of 16 and 32, respectively. For gradient magnitude annealing (Section 3.4), the values of γ follow a reverse sigmoid schedule, transforming the range $[-5, 5]$ with a sigmoid function and scaling it to $[0.01, 0.15]$. We filter gradients with an initial threshold $\eta_0 = 0.01$, decaying it exponentially by 0.99 after each rejection.

Baselines. We compare our method to state-of-the-art diffusion-based ones from three different categories: (1) instruction-guided (InstructPix2Pix or IP2P [6] and Hive [51]), (2) diffusion inversion-based (LEDITS++ [5]), and (3) score distillation (DDS [16] and CDS [30]).

We provide reference comparisons with task-specific supervised object insertion methods [45, 52] and rectified-flow-based methods [38, 42] in Appendix 10.3 and 10.4, as these tackle related tasks but are orthogonal to our contributions of stabilizing score distillation for general image editing in diffusion models.

Dataset. We use the MagicBrush test set [49], a standard benchmark for image editing featuring diverse types of edits. It contains 1,053 examples, each with (1) a source image $\mathbf{x}^{\text{src}}$, (2) its global description y^{src} , (3) a ground-truth target image $\mathbf{x}^{\text{tgt}}$, (4) its global description y^{tgt} , (5) an edit instruction y^{edit} , and (6) a local description of the edited region y^{local} . We use y^{edit} as input for instruction-guided models

and global descriptions y^{src} and y^{tgt} , which describe the *entire* image, to condition all other methods, including ours.

4.1. Human evaluation

We conducted a user study to assess the quality of image editing using 200 randomly selected samples from the MagicBrush test set [49]. For each sample, we presented a side-by-side comparison between our editing results and those of a state-of-the-art competitor. Each comparison was shown to 5 workers on Amazon Mechanical Turk, who were asked to choose their preferred image based on four criteria: (1) background preservation, (2) prompt fidelity, (3) quality of edited elements, and (4) overall preference. These criteria are similar to those proposed by EditVal [4]. As shown in Table 1, our method outperforms both instruction-based and global description-based competitors across all metrics. See Appendix 9.2 for details on the study design.

Method	Background	Prompt	Quality	Overall
IP2P [6]	33.5%	40.0%	36.5%	36.0%
HIVE [51]	47.0%	40.5%	45.0%	39.0%
LEDITS++ [5]	35.5%	33.0%	37.0%	35.0%
DDS [16]	43.5%	37.0%	38.0%	38.5%
CDS [30]	44.5%	40.0%	43.0%	42.0%
SBP [26]	61.7%	59.8%	61.5%	57.7%

Table 1. Percentage of times users preferred other methods over ours in 1-on-1 comparisons across different criteria.

4.2. Quantitative evaluation

Metrics. We evaluate two main aspects with five metrics: prompt fidelity (CLIP-T) and background preservation (CLIP-R, CLIP-AUC, L1*, and CLIP-I*). For prompt fidelity, we use CLIP-T, following [39, 49], which calculates the cosine similarity between the CLIP embeddings of the edited image $\mathbf{x}$ and the text prompt y^{local} , describing the local changes. For background preservation, our study (Appendix 9.3) shows that the L1, CLIP-I, DINO metrics used in prior work [39, 49], which are computed between the output and the single ground-truth edited image, are inherently biased: a method that does nothing to the input ranks first across the board, leading to misleading interpretations. To address this, we propose CLIP-R, CLIP-AUC, and improved versions of L1* and CLIP-I*. CLIP-R is defined as:

$$\text{CLIP-R} = \frac{\text{CosineSim}(\text{CLIP}(\mathbf{x}), \text{CLIP}(y^{\text{tgt}}))}{\text{CosineSim}(\text{CLIP}(\mathbf{x}^{\text{src}}), \text{CLIP}(y^{\text{tgt}}))}, \quad (5)$$

which quantifies how much more the edited image $\mathbf{x}$ conforms to the target prompt y^{tgt} than the input image $\mathbf{x}^{\text{src}}$ does. Since y^{tgt} describes entire images, methods are penalized for altering background elements specified in y^{tgt} . Because different inputs require varying degrees of modification to match y^{tgt} , we plot the ratio of edits whose

CLIP-R $> k$ for various thresholds $k \geq 1$ and compute the area under this curve as another metric, CLIP-AUC.

To address the shortcomings of L1 and CLIP-I for background preservation, we first ensure that edits from each method reach the same degree before computing scores. This is done by plotting the mean L1 and CLIP-I scores for edits with CLIP-R $> k$ for multiple k values and computing the area under curves (L_1^* and CLIP-I*). The integrals are computed for $k \in [1.0, 1.22]$, which excludes failed edits (CLIP-R < 1) and extends to the largest k that still produces at least 30 examples in any method for statistical analysis.

Results. As shown in Table 2, our method outperforms both existing zero-shot and supervised methods across most metrics, except for CLIP-I*. Since DDS and CDS struggle with object insertion, our improvements in Table 3 are more pronounced when evaluating only on such examples (see Appendix 9.1 for how we classify examples). Figure 8 indicates that at the same success rate, our results best align with the target caption y^{tgt} , which requires both strong textual alignment and background preservation. Our AUCs are also the largest, suggesting a better trade-off between the two aspects among all methods. This finding also concurs with the user study in Table 1.

Method	Time (mins)	CLIP-T $\uparrow$	CLIP-AUC $\uparrow$	L1* $\downarrow$	CLIP-I* $\uparrow$
Instruction-guided methods					
IP2P [6]	0.06	0.275	0.053	0.029	0.180
HIVE [51]	0.13	0.272	0.040	0.024	0.189
Global description-guided methods					
LEDITS++ [5]	0.13	0.279	0.067	0.022	0.182
DDS [16]	0.22	0.277	0.048	0.017	0.195
CDS [30]	0.62	0.272	0.034	0.016	0.197
SBP [26]	0.30	0.285	0.068	0.024	0.174
Ours	1.79	0.287	0.074	0.015	0.192

Table 2. Scores on MagicBrush of state-of-the-art methods and our method. The **best** and **second-best** scores are color-coded.

Method	CLIP-T $\uparrow$		CLIP-AUC $\uparrow$	
	All	Add	All	Add
DDS [16]	0.277(-3.6%)	0.266(-6.0%)	0.048(-35.1%)	0.043(-47.2%)
CDS [30]	0.272(-5.1%)	0.262(-7.7%)	0.034(-53.8%)	0.029(-64.4%)
SBP [26]	0.285(-0.6%)	0.281(-0.8%)	0.068(-8.6%)	0.069(-14.7%)
Ours	0.287	0.283	0.074	0.080

Table 3. Our method outperforms DDS, CDS, and SBP, with larger gains in object insertion tasks (Add), compared to all tasks (All).

4.3. Qualitative results

We present qualitative results of MagicBrush test inputs (Figure 7 and Appendix 10.1) and in-the-wild inputs (Figure 1, Figure 10, and Appendix 10.2). Compared to other general image editing techniques, our method better preserves the source background while more faithfully reflecting the target prompt for both large edits (e.g., dragon, pizza

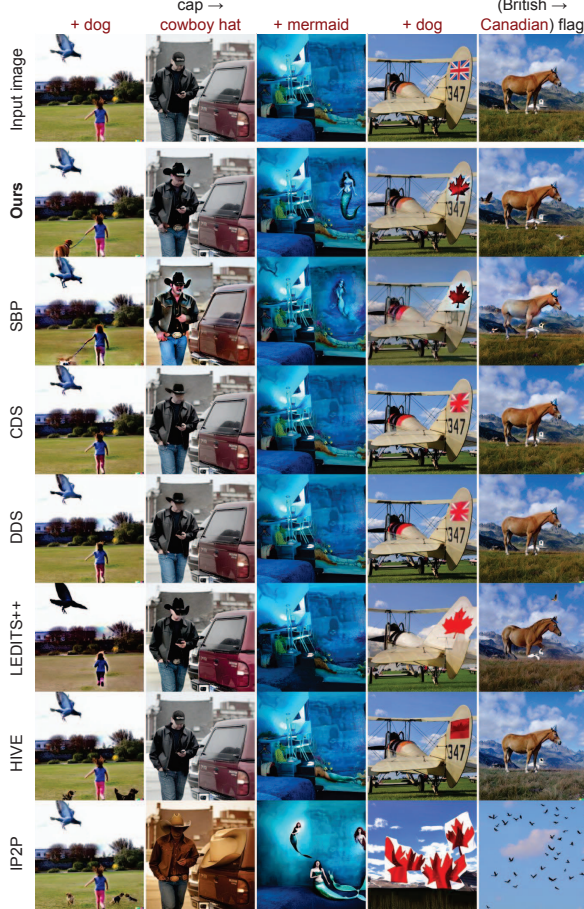


Figure 7. Qualitative results on MagicBrush dataset [49].

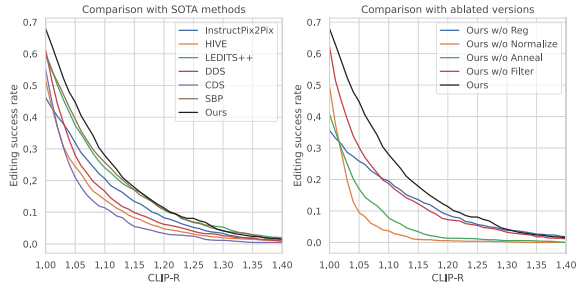


Figure 8. Editing success rate, defined as the ratio of edits with $\text{CLIP-R} > k$ at various thresholds $k \geq 1$. Our method achieves the most successful edits on MagicBrush [49].

toppings, lava) and subtle ones (e.g., cat’s eyes, sunglasses).

4.4. Ablation studies

We perform an ablation study on the MagicBrush dataset by removing (1) attention-based spatial regularization (setting $\lambda = 0$ and the mask $\mathbf{M}_k = 1$), (2) normalization (setting the denominator in Equation 4 to 1), (3) annealing (setting γ in Equation 4 to 1), and (4) filtering (setting the threshold for filtering gradients $\eta_0 = 0$).

Table 4 shows that our full method outperforms all ablated versions on CLIP-T. Without gradient filtering and

normalization, our method often fails to insert objects or adds incomplete ones, resulting in a lower CLIP-T score. Using a constant $\gamma = 1$ instead of annealing leads to unstable optimization, over-saturated outputs, and worse CLIP-T. Studies on the moving average in attention masks and other hyperparameters are in Appendices 7 and 8, respectively.

Method	CLIP-T $\uparrow$	CLIP-AUC $\uparrow$	L1* $\downarrow$	CLIP-I* $\uparrow \uparrow$
w/o Spatial Reg.	0.277	0.049	0.057	0.137
w/o Normalize	0.268	0.017	0.013	0.200
w/o Anneal ($\gamma = 1$)	0.265	0.025	0.021	0.185
w/o Filtering	0.279	0.053	0.015	0.195
Ours	0.287	0.074	0.015	0.192

Table 4. Ablations on MagicBrush. Our method best balances prompt fidelity and background preservation (CLIP-AUC).



Figure 9. Score distillation tends to be biased towards regions with existing visual cues, as they require less *effort* to modify.

4.5. Limitations and discussion

We highlight interesting failure cases of our method, and potentially score distillation in general, in Figure 9. These methods tend to favor minimal-effort regions, where visual cues for object formation already exist, leading to unnatural placements in some cases (see Appendix 11).

Additionally, our technique can be slow with gradient filtering, especially for challenging prompts that produce many problematic gradients. It also struggles with certain edits due to limited language understanding of diffusion models. However, using larger models with improved text understanding [35] can directly improve its performance. We also show in Appendix 14 that our proposed techniques can function as plug-and-play components and improve other distillation algorithms, such as DDS [16].

5. Conclusion

We introduce LUSD, an SDS-based method for text-guided image editing with an emphasis on object insertion. It features two new techniques that improve reliability in the face of input variation and randomness inherent to the optimization process: (1) attention-based spatial regularization to modulate gradients for background preservation, and (2) gradient filtering and normalization to mitigate counter-productive gradients. It outperforms other SOTA methods in prompt fidelity and has a higher rate of editing success while using a *single* set of hyperparameters for all inputs.

Acknowledgement: This research was supported by Research Fellowships from VISTEC, SCB public company limited, and PTT public company limited.

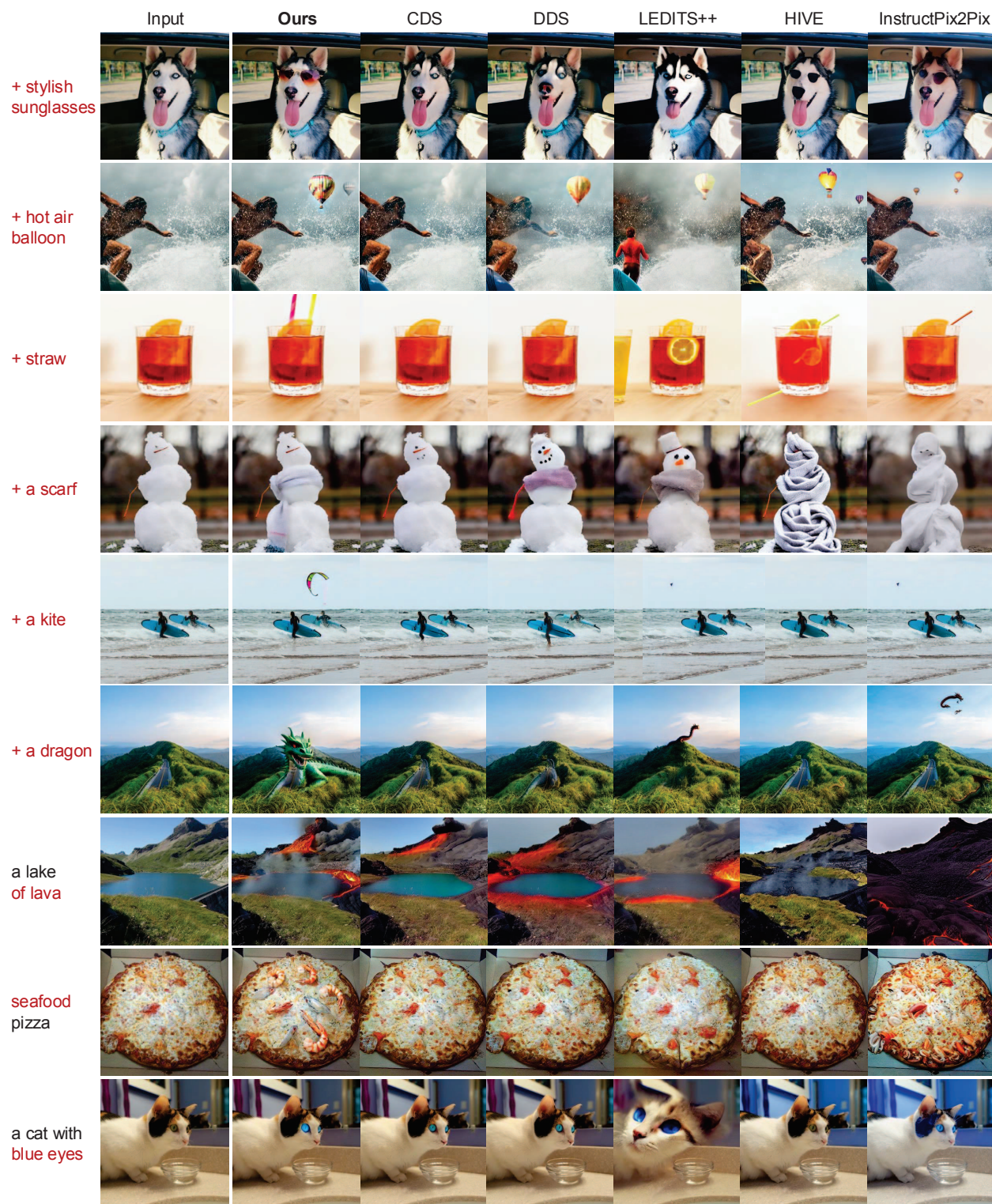


Figure 10. Qualitative results on various in-the-wild editing tasks. While state-of-the-art methods require instance-level hyperparameter tuning, our method successfully performs edits with a higher success rate using a *single* configuration. Hypertuning grids for results from state-of-the-art methods are provided in Appendix 12.

References

- [1] Thimo Alldieck, Nikos Kolotouros, and Cristian Sminchisescu. Score Distillation Sampling with Learned Manifold Corrective, 2024. [3](#)
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. [2](#)
- [3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics*, 42(4): 1–11, 2023. [2](#)
- [4] Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods, 2023. [6](#)
- [5] Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. 2023. [1](#), [2](#), [3](#), [5](#), [6](#), [4](#)
- [6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. [1](#), [2](#), [5](#), [6](#), [4](#)
- [7] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. [1](#), [3](#), [5](#), [6](#)
- [8] Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael S. Ryoo. Diffusion illusions: Hiding images in plain sight, 2023. [3](#)
- [9] 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/CVF International Conference on Computer Vision (ICCV)*, pages 22560–22570, 2023. [3](#)
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. [3](#)
- [11] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023. [4](#)
- [12] Sherry X Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. Tino-edit: Timestep and noise optimization for robust diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6337–6346, 2024. [2](#)
- [13] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance, 2022. [1](#), [2](#)
- [14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yan-nik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. [3](#)
- [15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. 2022. [1](#), [3](#)
- [16] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2328–2337, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [10](#)
- [17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. [3](#), [4](#)
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020. [1](#), [2](#), [3](#)
- [19] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly DDPM noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024. [2](#)
- [20] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation, 2023. [3](#)
- [21] Jeongsol Kim, Geon Yeong Park, and Jong Chul Ye. Dream-sampler: Unifying diffusion sampling and score distillation for image manipulation. *arXiv preprint arXiv:2403.11415*, 2024. [3](#), [4](#)
- [22] Juil Koo, Chanhoo Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, 2024. [3](#)
- [23] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. [4](#)
- [24] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Lucidreamer: Towards high-fidelity text-to-3d generation via interval score matching, 2023. [2](#)
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. [5](#)
- [26] David McAllister, Songwei Ge, Jia-Bin Huang, David W. Jacobs, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Rethinking Score Distillation as a Bridge Between Image Distributions, 2024. [2](#), [3](#), [4](#), [6](#), [5](#)
- [27] 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. [1](#), [2](#)
- [28] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models, 2023. [2](#)
- [29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models, 2022. [2](#)
- [30] Hyelin Nam, Gihyun Kwon, Geon Yeong Park, and Jong Chul Ye. Contrastive denoising score for text-guided latent diffusion image editing, 2024. [2](#), [3](#), [5](#), [6](#), [4](#), [10](#)

- [31] Quang Ho Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic dataset generation for pixel-level semantic segmentation. In *Proceedings of the Thirty-Seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 4, 5
- [32] OpenAI. Chatgpt: Language model for conversational ai. <https://chat.openai.com>, 2023. Accessed: 2024-11-15. 4
- [33] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020. 3
- [34] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation, 2023. 1, 3
- [35] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 7
- [36] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2, 3, 4
- [37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 3, 4, 2, 5
- [38] L Rout, Y Chen, N Ruiz, C Caramanis, S Shakkottai, and W Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *The Thirteenth International Conference on Learning Representations*, 2025. 3, 5
- [39] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks, 2023. 1, 3, 6
- [40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. 2020. 1, 2
- [41] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation, 2022. 3
- [42] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024. 3, 5
- [43] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J. Fleet, Radu Soricut, Jason Baldridge, Mohammad Norouzi, Peter Anderson, and William Chan. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting, 2023. 2
- [44] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation, 2023. 3
- [45] Navneet Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. Paint by inpaint: Learning to add image objects by removing them first. *arXiv preprint arXiv:2404.18212*, 2024. 5
- [46] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model, 2022. 2
- [47] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. *arXiv preprint arXiv:2211.13227*, 2022. 2, 3
- [48] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023. 3
- [49] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023. 2, 5, 6, 7, 3, 4, 8, 13, 14
- [50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2
- [51] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, Caiming Xiong, and Ran Xu. Hive: Harnessing human feedback for instructional visual editing, 2024. 1, 2, 3, 5, 6, 4
- [52] Lirui Zhao, Tianshuo Yang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Rongrong Ji. Diffree: Text-guided shape free object inpainting with diffusion model. *arXiv preprint arXiv:2407.16982*, 2024. 5
- [53] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting, 2024. 2