Supplementary Material
High-Fidelity Guided Image Synthesis with Latent Diffusion Models

Jaskirat Singh[1] Stephen Gould[1,2] Liang Zheng[1,2]
[1]The Australian National University [2]Australian Centre for Robotic Vision
{jaskirat.singh, stephen.gould, liang.zheng}@anu.edu.au

A. Additional Results

In this section, we provide additional results which could not be included in the main paper due to space constraints. In particular, we note that baseline methods like SDEdit [10] can often be run using different values of the hyperparameter t_0. We therefore provide additional results comparing the performance of SDEdit at different $t_0 \in [0, 1]$ (refer Sec. A.1). Additionally, we introduce some custom baselines (which could be used for improving the realism of final image outputs) and show results comparing their output performance with our approach (refer Sec. A.2).

A.1. Additional Comparisons with SDEdit

Recall, given a stroke painting y, SDEdit [10] follows an inversion-based approach for performing guided image synthesis. In particular, the generative prior is introduced by first passing the painting y through the forward diffusion pass $y \rightarrow y_{t_0}$ [7, 15], and then performing reverse diffusion $y_{t_0} \rightarrow y_0$ to get the output image $x = y_0$. Due to space constraints, we primarily use the standard hyperparameter value of $t_0 = 0.8$ in the main paper. In this section, we provide additional results which comprehensively compare our approach with SDEdit [10] under changing values of t_0.

Qualitative Comparisons. Results are shown in Fig. 2, 3. We observe that for lower values of t_0, SDEdit generates outputs which though highly faithful to the reference painting, lack details and represent simplistic representations of the target image. Increasing the value of hyperparameter t_0 helps improve realism but the outputs become less and less faithful with the reference image. In contrast, the proposed approach leads to outputs which are both *faithful* to the reference painting as well as exhibit high *realism w.r.t* the target domain (generated only using the text prompt).

Quantitative Comparisons. In addition to qualitative results, we also report quantitative results by analysing the relationship between the *faithfulness* $\mathcal{F}$ and *realism* $\mathcal{R}$ metrics (refer Sec. 4.1 of main paper), under changing values hyperparameter t_0. Results are shown in Fig. 1. We observe that as compared to prior works, our method provides the best tradeoff generating *realistic* outputs and maintaining *faithfulness* with the provided reference painting.

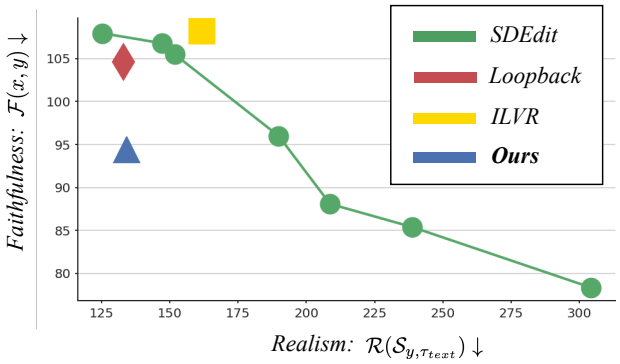

Figure 1. ***Visualizing faithfulness-realism tradeoff.*** We analyse the tradeoff between faithfulness-realism distances for different methods (note that lower is better for both metrics). We observe that as compared to prior works, our method provides the best tradeoff between generating realistic outputs and maintaining faithfulness with the provided reference painting.

A.2. Comparison with Custom Baselines

In this section, we introduce some custom methods (as baselines) for increasing the realism of generated outputs with SDEdit [10], and then compare the output performance for the same with our approach. In particular, we show additional comparisons with the following custom baselines,

- ***Attention Re-weighting (AttnRW)*** [3] wherein the realism *w.r.t* the target domain is enhanced by increasing the attention weighting for the corresponding domain specific text tokens (*e.g.* photo, painting *etc.*). For instance, if the text prompt says *"a photo of a tree"*, then we aim to increase the realism of the generated outputs by increasing the weightage of the cross-attention maps corresponding to the the word *"photo"* [3]. Results are shown in Fig. 4. We observe that while increasing the weightage of domain specific text tokens (*e.g.* photo, painting *etc.*) helps improve the realism of the output images to some extent, the final images still lack details and certain blurry re-

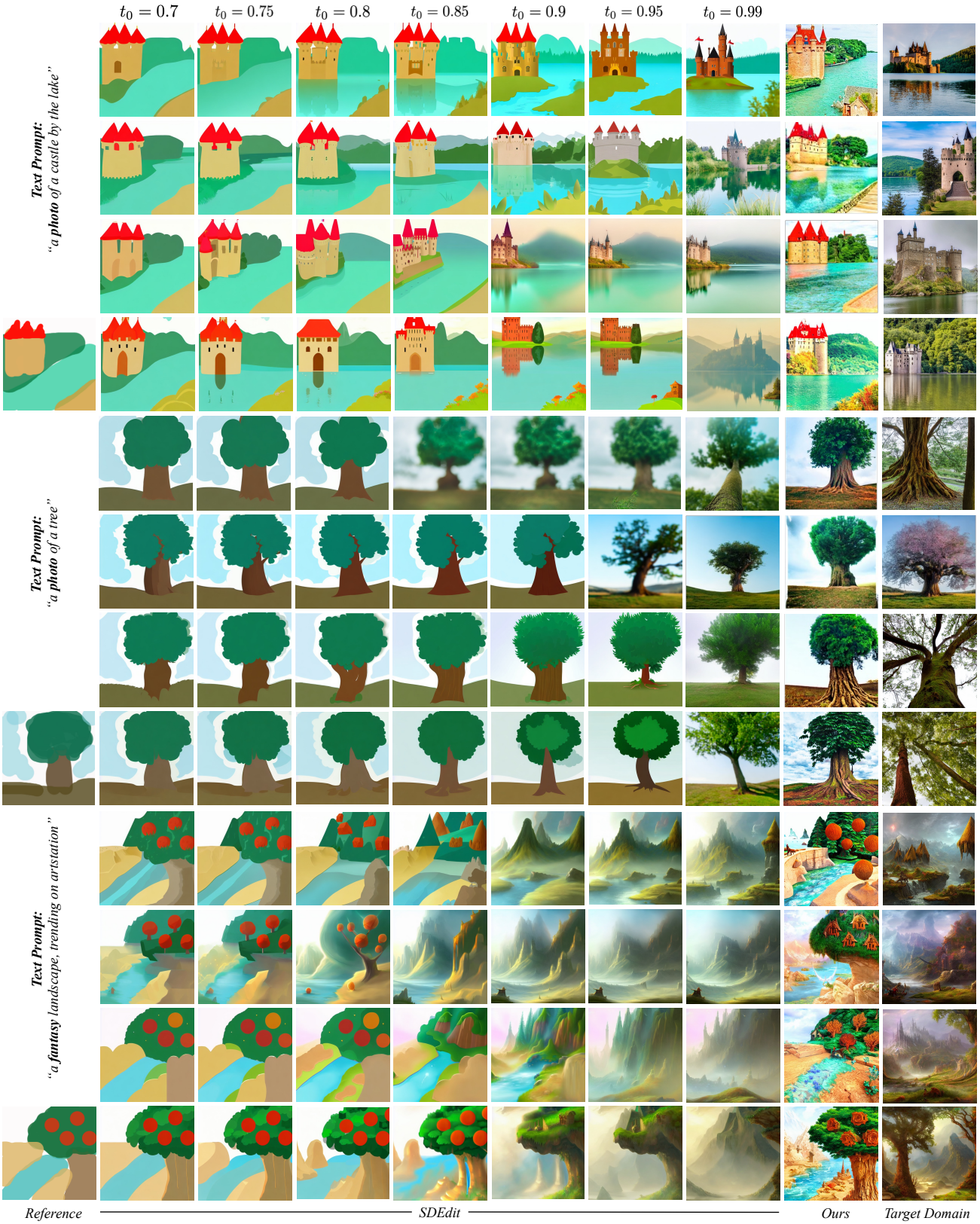

Figure 2. ***Additional comparisons.*** We provide comprehensive comparisons with SDEdit [10] under changing value of hyperparameter t_0. We find that SDEdit [10] either generates faithful but cartoon-like outputs for low t_0, or, generates realistic but unfaithful outputs at high t_0. In contrast, our approach leads to outputs which are both realistic (*w.r.t* the target domain) as well as faithful (to the provided reference).

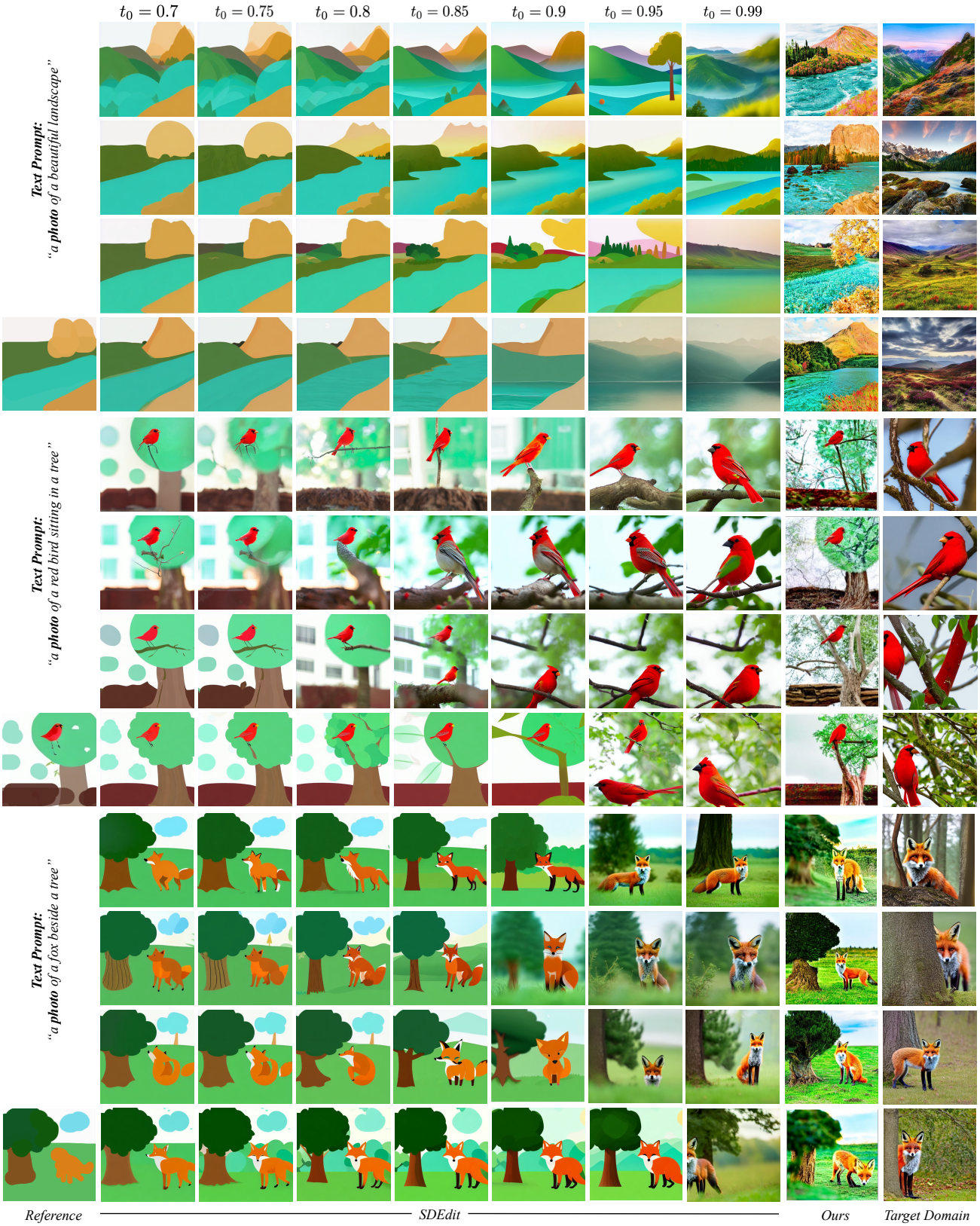

Figure 3. ***Additional comparisons.*** We provide comprehensive comparisons with SDEdit [10] under changing value of hyperparameter t_0. We find that SDEdit [10] either generates faithful but cartoon-like outputs for low t_0, or, generates realistic but unfaithful outputs at high t_0. In contrast, our approach leads to outputs which are both realistic (*w.r.t* the target domain) as well as faithful (to the provided reference).

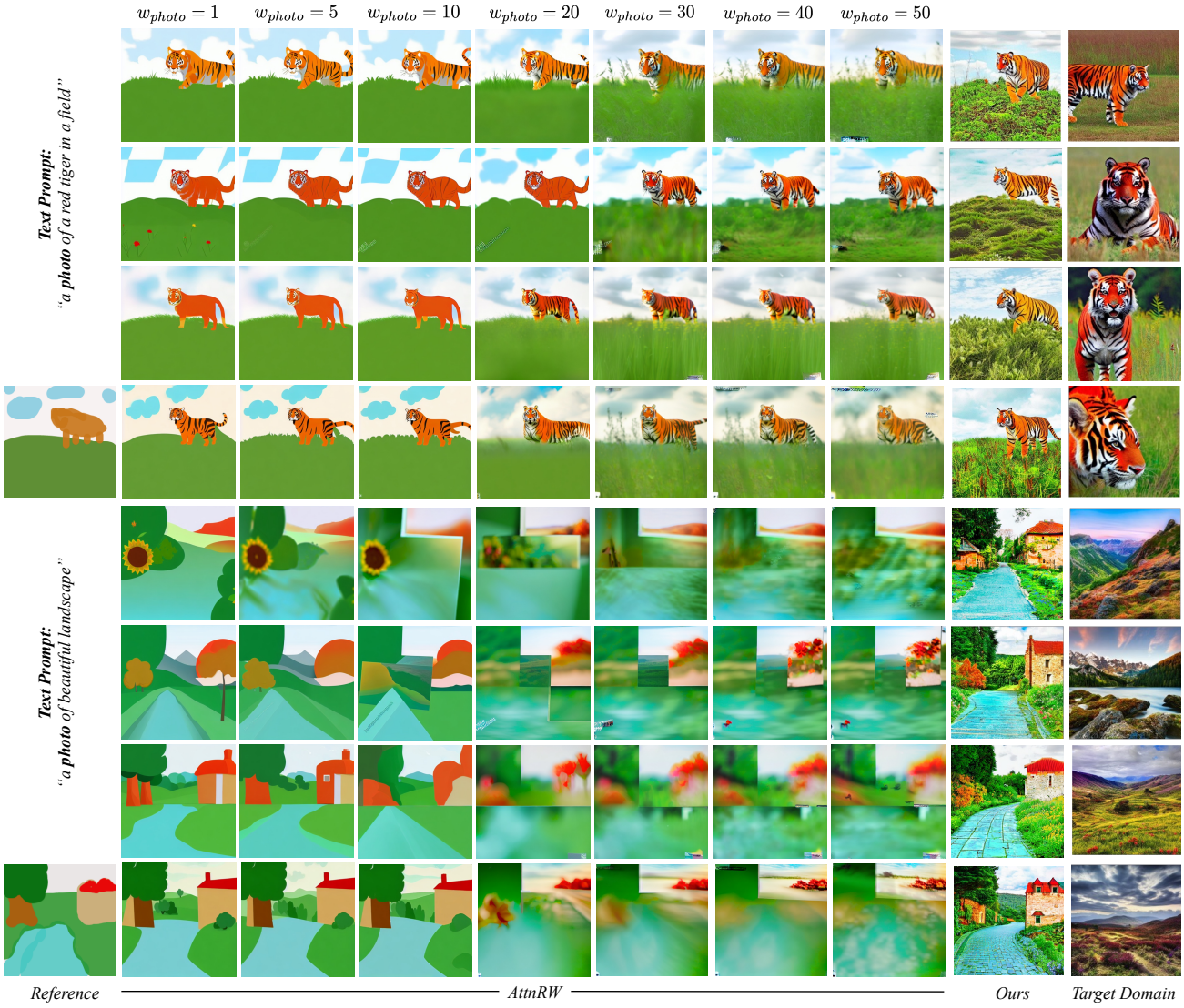

Figure 4. ***Comparison with Custom Baselines - AttnRW*** *[3].* We compare the performance of our method with the Attention Reweighting (AttnRW) approach for increasing realism *w.r.t* the target domain. We find that increasing the weight of cross attention maps corresponding to the domain-specific text tokens (*e.g*. photo in above), leads to improved realism of the generated outputs. However, we note that certain blurry details persist *e.g*. grass in row 1-4. Also, the increase in realism is accompanied by some image artifacts *e.g*. blotched image regions in row 1-4, image in image artifacts in row 4-8 *etc*. In contrast, our approach improves output realism in a more coherent manner.

gions still persist (*e.g*. grass in row-1). Furthermore, the increase in realism is accompanied by some image artifacts *e.g*. blotched image regions in row 1-4, image-in-image artifacts in row 4-8 *etc*. In contrast, we find that our method provides a more practical approach for increasing the output realism in a semantically coherent manner.

- ***Increasing Classifier Guidance Scale*** [5], wherein we attempt to increase the realism of the SDEdit [10] outputs by increasing the scale of classifier free guidance used during the reverse diffusion process. Results are shown in Fig. 5. We observe that while increasing the scale of classifier free guidance improves the level of detail in the generated images, the final outputs still resemble cartoon-like or simplistic representations of the target domain. Furthermore, we also note that our approach can also benefit from the increase in guidance scale in order to increase the level of fine-grain detail in the output images.

B. Experiment Details

B.1. Implementation Details

In this section, we provide further details for the implementation of our approach as well as other baselines used while reporting results in the main paper.

Ours. We use publicly available text-conditioned latent diffusion models [11,16] for implementing the purposed ap-

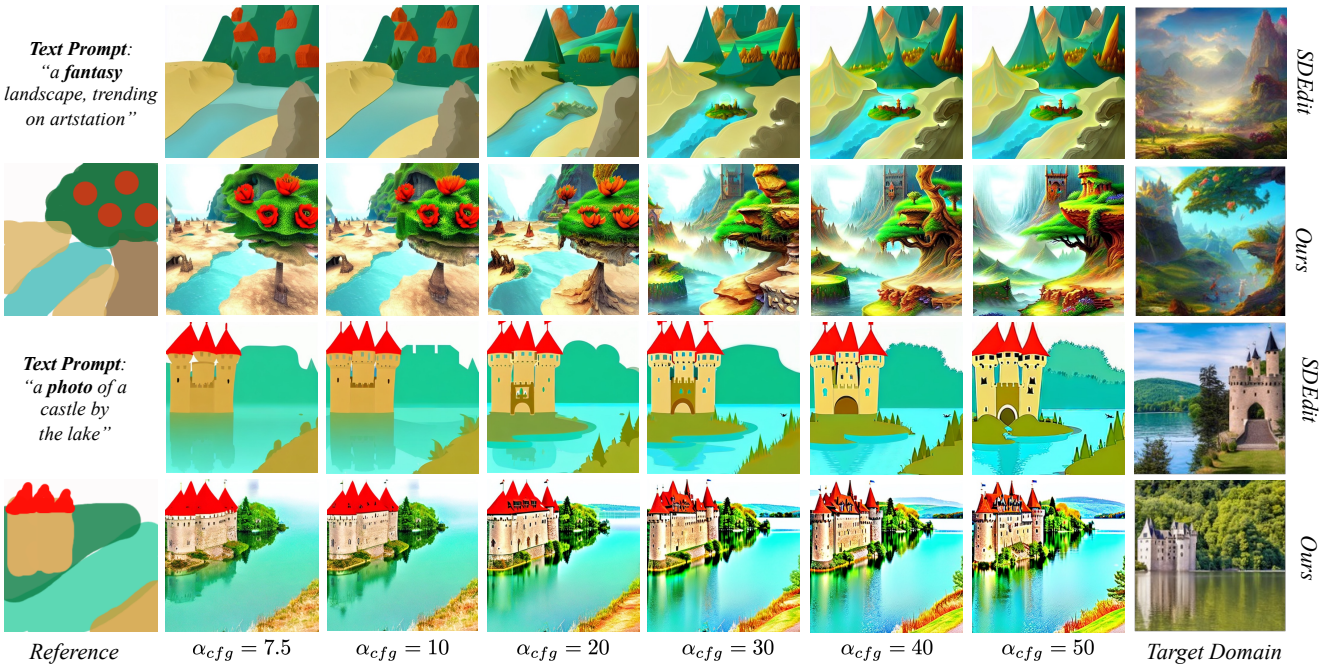

Figure 5. ***Comparison with Custom Baselines - CFG*** [5]. We analyse the impact of increasing the classifier-free guidance scale α_{cfg} on outputs generated using SDEdit [10] and our method. We find that while increasing the value of α_{cfg} leads to increase in level of details, the final outputs still represent simplistic representations of the target domain (row-3). Furthermore, as the value of α_{cfg} is increased, the faithfulness with respect to the reference painting is compromised (*e.g.* red regions in row-1).

proach in the main paper. The constrained optimization is performed using gradient descent with the Adam [8] optimizer and number of gradient steps $N_{grad} \in [20, 60]$. While several formulations of the distance measure $\mathcal{L}$ and painting function f are possible (refer Sec. C), we find that simply approximating the function $\mathcal{L}$ using mean squared distance and f as a convolution operation with a gaussian kernel seems to give the fastest inference time performance with our method. For consistency with prior works, we use the non-differentiable painting function from SDEdit [10] while reporting quantitative results. All results are reported using the DDIM sampling [15] with 50 inference steps for performing the reverse diffusion process.

SDEdit [10]. We use the standard image-to-image pipeline from the open-source *diffusers* library [16] for reporting results for *SDEdit* [10] with different values of hyperparameter $t_0 \in [0, 1]$. Similar to our method, all results are reported at 512×512 resolution using DDIM sampling [15] with 50 inference steps for performing the reverse diffusion process. Unless otherwise specified, a classifier-free guidance scale [5] of $\alpha_{cfg} = 7.5$ is used for all experiments.

SDEdit + Loopback [1]. We use the previously described SDEdit implementation and iteratively reperform guided synthesis with the previous diffusion outputs to improve realism of the generated outputs. In particular, we use $N_{iter} = 4$ iterations for the iterative process. Also, similar to [1], in order to increase the realism of generated outputs with each iteration, the hyperparameter t_0 is updated as,

$$t_0^{n+1} \leftarrow min(t_0^n \cdot k, 1.0), \quad k \in [1.0, 1.1] \tag{1}$$

where $n \in [1, N_{iter}]$ is the iteration number. Unless otherwise specified, we use the standard hyperparameter selection of $k = 1.05$ and $t_0^{n=1} = 0.8$ for our experiments.

ILVR [2]. The original ILVR [2] algorithm was proposed for iterative refinement with diffusion models in pixel space. We adapt the ILVR implementation for inference with latent diffusion models [11] for the purposes of this paper. In particular given a reference painting y, the original ILVR algorithm modifies the diffusion output x_t (in pixel space) at any timestep t during reverse diffusion process as,

$$\tilde{x}_t = \phi_N(y_t) + x_t - \phi_N(x_t), \quad y_t \sim q(y_t \mid y) \tag{2}$$

where $q(y_t \mid y)$ represents the forward diffusion process from $y \rightarrow y_t$, $\phi_N(.)$ is a low pass filter achieved by scaling down the image by a factor of N and then upsampling it back to the original dimensions. Assuming a latent diffusion model with encoder $\mathcal{E}$ and decoder $\mathcal{D}$, we simply adapt the above update in latent space as follows,

$$x_t = \mathcal{D}(z_t) \tag{3}$$

$$z_y = \mathcal{E}(y), \; z_{y_t} \sim q(z_{y_t} \mid z_y) \tag{4}$$

$$\tilde{x}_t = \phi_N(y_t) + x_t - \phi_N(x_t), \quad y_t = \mathcal{D}(z_{y_t}) \tag{5}$$

$$\tilde{z}_t = \mathcal{E}(\tilde{x}_t) \tag{6}$$

where Eq. 3, 6 map the latent features z_t to pixel space x_t, and vice-versa. Eq. 4 computes y_t from y by first mapping y to z_y, computing the forward diffusion $z_y \rightarrow z_{y_t}$ and then reverting back z_{y_t} to y_t. Finally, Eq. 5 is simply the original update rule from ILVR algorithm [2]. A hyperparmeter value of $N = 4$ is used while reporting results.

B.2. Quantitative Experiments

Data Collection. Since there is no predefined dataset for guided image synthesis with user-scribbles and text prompts, we create our own dataset for reporting quantitative results. In particular, we first collect a set of 100 stroke painting and text prompt pairs from diverse data modalities with the help of actual human users. We then augment the collected data using a prompt-engineering approach to increase the diversity of the collected data pairs. In particular, the text prompt for each data-pair is modified in order to in order to replace the domain specific text words (*e.g.* photo, painting) with pre-designed target domain templates, while keeping the underlying content the same. During prompt engineering, the target domain template is chosen randomly from *['photo','watercolor painting', 'Vincent Van Gogh painting','children drawing','high resolution disney scene', 'high resolution anime scene', 'fantasy scene','colored pencil sketch']*. For each data pair, we then sample four random guided image synthesis outputs for each baseline and our method. The resulting dataset consists of 800 (painting, text-prompt) pairs and 3200 overall samples from diverse data modalities for final method evaluation.

Quantitative Metrics. In order to evaluate the performance of our approach, we introduce two metrics for measuring the *faithfulness* of the output *w.r.t* the reference painting, and the *realism* of the generated samples *w.r.t* the target domain (specified through text-only conditioning). In particular, given an input painting y and output real image prediction x, we define faithfulness distance $\mathcal{F}(x, y)$ as,

$$\mathcal{F}(x, y) = \mathcal{L}_2(f(x), y) \tag{7}$$

where $f(.)$ is the painting function. Thus an output image x is said to have high faithfulness with the given painting y if upon painting the final output x we get a painting $\tilde{y} = f(x)$ which is similar to the original target painting y (Fig. 6).

The painting function f is implemented using the human stroke-simulation algorithm from SDEdit [10]. In particular, given an 256×256 input image, the output painting is computed by first passing the image through a median filter with kernel size 23, and then perform color quantization to reduce the number of colors to 20 using an adaptive palette.

Similarly, given a set of output data samples $\mathcal{S}(y, \tau_{text})$ conditioned on both painting y and text τ_{text}, and, $\mathcal{S}(\tau_{text})$ conditioned only on the text, the *realism* $\mathcal{R}$ is defined as,

$$\mathcal{R}(\mathcal{S}(y, \tau_{text})) = FID\left(\mathcal{S}(y, \tau_{text}), \mathcal{S}(\tau_{text})\right) \tag{8}$$

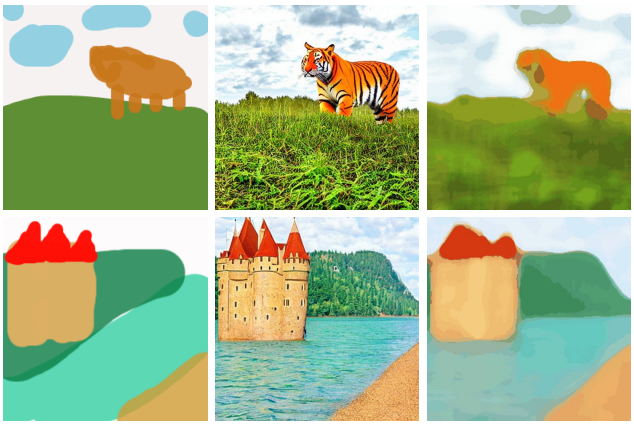

Figure 6. Visualizing input painting y, output x and painted reconstruction $\tilde{y} = f(x)$. The goal is to generate an output x which is realistic and for which painting loss $\mathcal{L}_2(f(x), y)$ is minimized.

where FID represents the Fisher inception distance [4].

Please note that while the above defined *realism* distance measure $\mathcal{R}$ captures the realism with respect to the target domain, we expect the computed FID scores to be higher than those expected of unconditioned image outputs. This is because while the proposed method generates outputs which seem realistic to human eyes, the variance of output distribution is significantly lower than that of real images. The decreased variance in output images occurs simply because the layout and color composition are predominantly fixed as a result of additional conditioning on the stroke painting y. In contrast, natural images or images conditioned only on the text prompt have a much higher diversity in terms of scene layout and the overall color composition. We therefore try to overcome of lack of diversity in generated image outputs by performing random data augmentations (random horizontal flip and random resized crop of size 448×448 on a 512×512 image) while computing the final realism scores across different methods [1].

Human User Study. In addition to reporting quantitative results using the above defined measures for *faithfulness* and *realism*, similar to [10], we also perform a human user study wherein the *realism* and the overall satisfaction score (*faithfulness* + *realism*) are evaluated by actual human users. For the *realism* scores, given an input text prompt (with target domain τ_{domain} *e.g.* τ_{domain} = '*photo*') and sample images conditioned only on the text prompt, the participants were shown a pair of image generation outputs comparing our method with prior works. For each pair, the human subject is then asked to select the output image which is more realistic with respect to

[1] Note that while this helps increase the diversity in scene layout the diversity in color composition is still lower than that of real images or image outputs conditioned only on the text prompt.

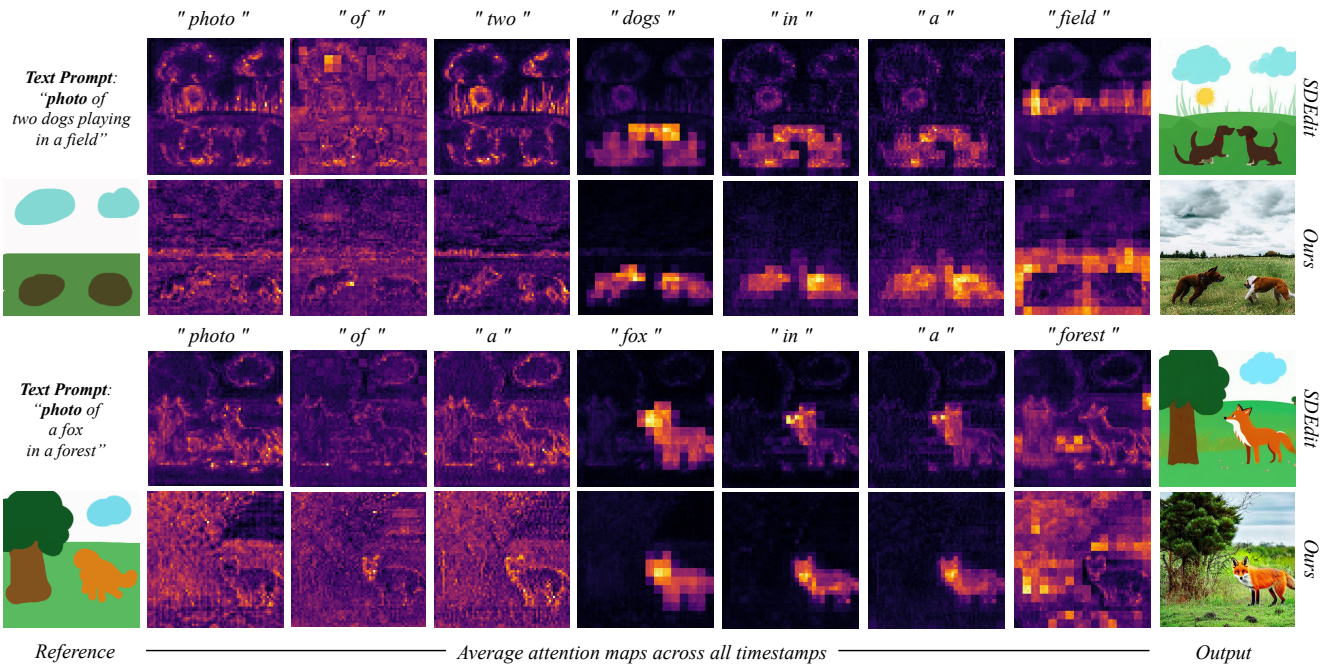

Figure 7. ***Visualizing the effect of GradOP on cross attention maps.*** We analyse the effect of our approach on the cross-attention maps generated during the reverse diffusion process. We find that our method leads to cross-attention outputs which help the model pay better attention to desired image areas in the reference painting. For instance, in the first example, the cross-attention features show high overlap with the desired *dog* and *field* regions. In contrast, the cross attention maps from SDEdit [10] reveal that the model is not paying adequate attention to the desired image areas (*e.g. field* in row-1, *tree and forest* in row-3) while generating the final output.

the target domain (τ_{domain}). Similarly, for computing the overall satisfaction scores, given an input stroke painting, text prompt and sample images conditioned only on the text prompt, the participants were shown a pair of image generation outputs comparing our method with prior works. For each pair, the instruction is: "Given the input painting and text prompt, how would you imagine this image to look like in reality? Your selection should be based on how realistic and less blurry the image is (please check level of details), consistency with the target domain (τ_{domain}) and whether it is faithful with the reference painting in terms of scene layout, color composition". For each task (*e.g.* computing overall satisfaction score), the collected data samples (discussed above) were divided among 50 human participants, who were given an unlimited time in order to ensure high quality of the final results. Additionally, in order to remove data noise, we use a repeated comparison (control seed) for each user. Responses of users who answer differently to this repeated seed are discarded while reporting the final results.

C. Method Analysis: Continued

C.1. Effect of GradOP on Cross Attention Maps

As shown by Hertz *et al*. [3] and our results, the cross-attention maps corresponding to different words in the input text prompt play a key role in deciding the overall semantic contents of the final image output. In this section, we try to analyse how the proposed approach leads to more realistic image content generation by analysing the average cross-attention maps generated while performing the reverse diffusion process with SDEdit [10] and our method.

Results are shown in Fig. 7. We find that our method leads to cross-attention outputs which help the model pay better attention to desired image areas in the reference painting. For instance, in the first example, the cross-attention features show high overlap with the desired *dog* and *field* regions. In contrast, the cross attention maps from SDEdit [10] reveal that the model is not paying adequate attention to the some desired image areas (*e.g. field* in row-1, *forest* in row-3) while generating the final output.

C.2. Semantic Control without Painting Guidance

Recall that in addition to performing *high-fidelity* guided image synthesis, we also show that by simply defining a cross attention based correspondence between the input text tokens and the user painting, it is possible to control the semantics of different image regions without the need for any semantic segmentation based conditional training. In this section, we analyse whether similar semantic control is possible without having additional guidance through a stroke painting. In particular, we wish to analyse if such fine-grain control is only possible while providing additional guidance

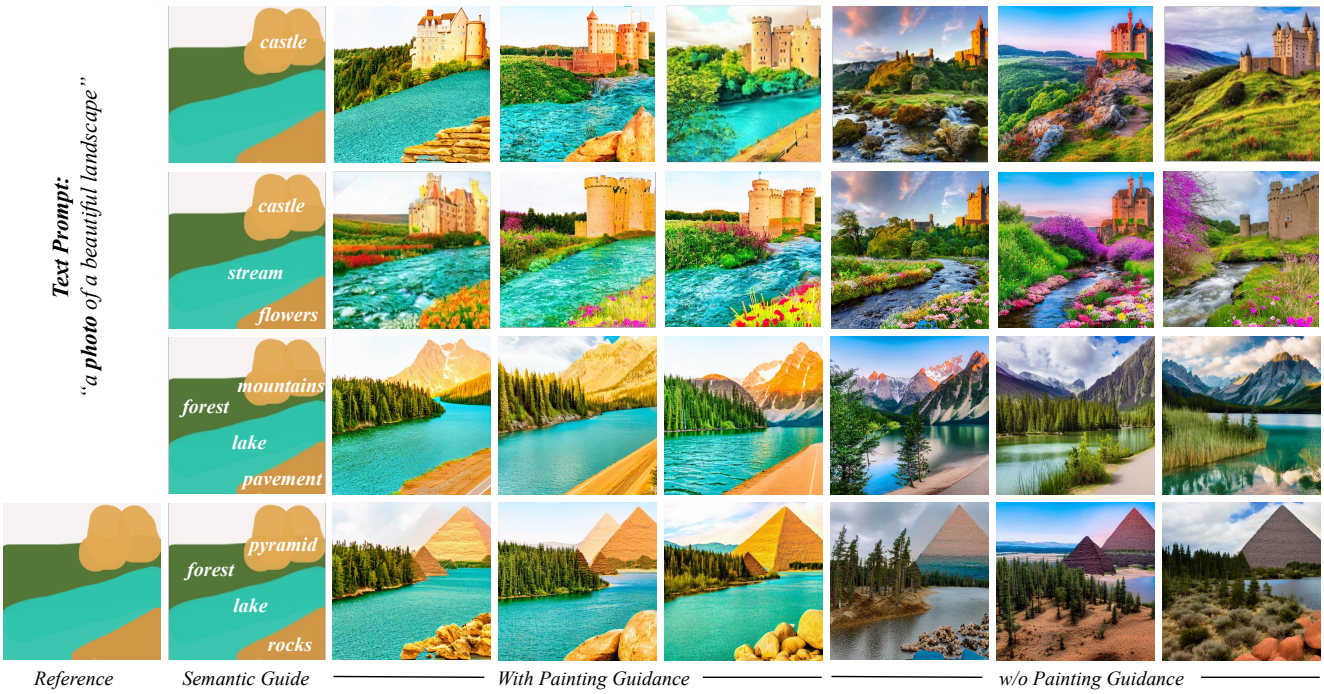

Figure 8. ***Analysing role of painting guidance in semantic control.*** We analyse the effect of using an underlying reference painting as guidance in controlling the semantics of different image areas using cross-attention based correspondence approach presented in the main paper (refer Sec. 3.3 in main paper). We find that additional guidance using reference stroke painting helps the user gain much accurate control over the semantics of different image regions (*e.g.* lake in row-3,4, mountains in row-3, rocks, forest in row-4 *etc.*).

through the reference stroke painting?

To answer this question, we compare the outputs generated through semantic control with and without using a reference painting for the guided synthesis process. Results are shown in Fig. 8. We observe that while for it is feasible to define the semantics of one or two parts of the image accurately using *cross-attention* correspondence, the performance decreases as the number of semantic labels increases (*e.g.* lake in row-3,4, mountains in row-3, rocks, forest in row-4 *etc.*). In contrast, we find that the use of a reference painting results in much better control over the semantics of different image regions. We believe that the same is because the use of a reference painting sets up a generic semantic structure for the output image which can then be easily refined by defining a cross-attention based correspondence. For instance, in row-4 of Fig. 8, adding the blue strokes for lake region sets up a semantic prior which constrains the inference of output semantics to semantic categories like river, lake, sea, stream, blue-green grass, blue pavement *etc.* The use of semantic correspondence then helps refine these output semantics to what is actually desired by the user. In contrast, without stroke guidance, the initial semantics for lake region could me much more diverse (*e.g.* sand, rocky terrain in row-4), and thereby more challenging to refine through the proposed semantic correspondence strategy.

C.3. Inference Time Analysis

We report a comparison of the average inference times required for each output image in Tab. 1. All results are reported using the DDIM sampling [15] with 50 inference steps, on a single Nvidia RTX 3090 GPU.

Method	Inference Time (s)	
	w/o mixed precision	*with mixed precision*
SDEdit [10]	6.32 s	4.45 s
Loopback [1]	27.2 s	20.46 s
ILVR [2]	8.24 s	6.17 s
GradOP **(Ours)**	20.1 s	15.8 s
GradOP+ **(Ours)**	12.3 s	8.86 s

Table 1. ***Inference time analysis***. Comparing inference time required for generating each output image for different methods. All results are reported with DDIM sampling and 50 inference steps.

C.4. Variation in Painting Function

Please recall that a key requirement for solving the proposed constrained optimization in Sec. 3 is to define a differentiable painting function f, which provides a good approximation for *"how a human would paint a given image with coarse user-scribbles"*. In this section, we therefore look at some possible formulations for obtaining an approx-

Figure 9. Analysing performance for different differentiable approximations of painting function f. We find that while using a more accurate painting function [10] (Col-2) leads to slightly more details (*e.g.* notice the gradient of the grass regions in row-1, detailed shadows of the castle and island in row-2), in practice more simpler approximations (*e.g.* Gaussian Blur) also produces highly realistic outputs while allowing for much faster inference times.

imation of the painting function in a differentiable manner[2], and compare the corresponding output results.

Painting Function Formulation. In particular, we consider three main formulations for constructing a differentiable painting function f, 1) *Median Filter + Color Quantization*, wherein we implement a differentiable approximation of the human-stroke simulation algorithm in [10]. In particular, given a reference painting y and output x, we first pass x through a median filter of size 23. We then pass the output of the last step through a differentiable color quantization function which maps the image pixels to their nearest rgb value in the painting y (that is, we are performing color quantization *w.r.t* the palette of the reference painting.) 2) *Median Filter* wherein we use the median filter alone for approximating the painting function, and 3) *Gaussian Blur* wherein approximate the painting function through a convolution operation with a Gaussian kernel (size 31 and σ=7).

Results are shown in Fig. 9. We observe that while the use of a more accurate human-stroke simulation function from [10] allows for the generation of slightly more detailed outputs (*e.g.* notice the gradient of the grass regions in row-1, detailed shadows of the castle and island in row-2), it increases the overall inference time required for the proposed gradient descent optimization (40.7s on *GradOP+*). In contrast, we find that using much more simpler approximations (*e.g.* Median Filter, Gaussian Blur) for the painting function also produces highly realistic outputs while allowing for much faster inference times (8.86s, 14.1s on *GradOP+* for Gaussian Blur and Median Filter respectively).

[2]Please note that while several more advanced formulations for designing the autonomous painting function are possible [6,9,12–14,17], they are usually non-differentiable. In this paper, we primarily limit the choice of painting functions to differentiable functions in order to allow for gradient descent based optimization with the proposed GradOP/GradOP+ methods.

References

[1] AUTOMATIC1111. Stable-diffusion-webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui, 2022. 5, 8

[2] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 5, 6, 8

[3] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1, 4, 7

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4, 5

[6] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8709–8718, 2019. 9

[7] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2435, 2022. 1

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. *arXiv preprint arXiv:2108.03798*, 2021. 9

[10] 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, 3, 4, 5, 6, 7, 8, 9

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 4, 5

[12] Jaskirat Singh, Cameron Smith, Jose Echevarria, and Liang Zheng. Intelli-paint: Towards developing human-like painting agents. In *European conference on computer vision*. Springer, 2022. 9

[13] Jaskirat Singh and Liang Zheng. Combining semantic guidance and deep reinforcement learning for generating human level paintings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 9

[14] Jaskirat Singh, Liang Zheng, Cameron Smith, and Jose Echevarria. Paint2pix: Interactive painting based progressive image synthesis and editing. In *European conference on computer vision*. Springer, 2022. 9

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 5, 8

[16] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4, 5

[17] Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15689–15698, 2021. 9