Watch Your Steps: Local Image and Scene Editing by Text Instructions

Ashkan Mirzaei[1,2] Tristan Aumentado-Armstrong[1,3,4]
Marcus A. Brubaker[1,3,4] Jonathan Kelly[2] Alex Levinshtein[1]
Konstantinos G. Derpanis[1,3,4] Igor Gilitschenski[2]

[1]Samsung AI Centre Toronto [2]University of Toronto [3]York University
[4]Vector Institute for AI

Abstract. The success of denoising diffusion models in generating and editing images has sparked interest in using diffusion models for editing 3D scenes represented via neural radiance fields (NeRFs). However, current 3D editing methods lack a way to both pinpoint the edit location and limit changes to the desired volumetric region. Consequently, these methods often over-edit, altering irrelevant parts of the scene. We introduce a new task, 3D edit localization, to automatically identify the relevant region for an editing task and restrict the edit accordingly. To achieve this goal, we initially tackle 2D edit localization, and then lift it to multiple views to address the 3D localization challenge. For 2D localization, we leverage InstructPix2Pix (IP2P) and identify the discrepancy between IP2P predictions with and without the instruction. We refer to this discrepancy as the relevance map. The relevance map conveys the importance of changing each pixel to achieve an edit, and guides downstream modifications, ensuring that pixels irrelevant to the edit remain unchanged. With the relevance maps of multiview posed images, we can define the *relevance field*, defining the 3D region within which modifications should be made. This enables us to improve the quality of text-guided 3D NeRF scene editing, by performing iterative updates on the training views, guided by renders from the relevance field. Our method achieves state-of-the-art performance on both NeRF and image editing tasks. We will make the code available.

Keywords: Generative models · 3D editing · Neural radiance fields

1 Introduction

As generative models improve, automated editing of various standard media (e.g., images [6,11,20,41,43] and videos [7,17,85]) with such models has become increasingly important and widespread. Further, new forms of *3D* media, such as neural radiance fields [46] (NeRFs) are increasingly accessible [9,51,58] and popular as an intuitive visualization modality; thus, editing NeRFs has also begun receiving significant attention [15,18,47,48]. However, editing with learned

Fig. 1: Overview of the relevance map calculation (left inset), and relevance-guided sample outputs on image (top-right inset) and neural radiance field (NeRF; bottom-right inset) editing. Given an image or a NeRF, our goal is to change the input according to a textual instruction. The relevance map is the disagreement between noise predictions with and without the instruction. For both image and scene editing, the relevance map confines the changes to the most relevant regions, according to the edit text.

models is often an ill-posed problem, particularly when the only given information is textual semantics. For instance, the command "add cherry blossoms" does not specify what kind, where in the image, or to what extent (e.g., across the whole image vs. on one tree) the edit should occur. Assuming a viewpoint-based editing algorithm, even greater difficulty is incurred in 3D, as the *inconsistency* of the edits across views can now induce additional problems. We argue that a notion of *minimal edit* can improve the controllability of such tasks. That is, when ambiguity is present, we should prefer parsimonious edits, where the fewest set of changes that still satisfy a desired edit are chosen.

Current diffusion-based image editors generally lack a mechanism to automatically localize the edit regions. The closest methods either ask users for a mask [41], rely on the global information kept in a noisy input as a starting point [43], or condition the denoiser on the input [6]. Nevertheless, all of these methods tend to over-edit, whether in 2D [6,11] or 3D [18]. Recently, DiffEdit [11] proposed using the difference between the noise predictions conditioned on captions to localize image edits, but it is slow, due to *denoising diffusion implicit model (DDIM)* [68] inversion, and it requires both input and output captions (while DiffEdit can technically be used with only the output caption, as discussed in their paper and shown in our experiments, omitting the input caption results in significantly worse edits). Further, its edits attempt to use a standard generative model to perform image translation. In contrast, the recent work Instruct-Pix2Pix (IP2P) [6], specialized for the task, has proven easier to use and more effective.

In this paper, we present an approach to *automatic localization* of an edit based on a single instruction (cf. DiffEdit [11], which requires input and output captions), acting as a controllable prior on its spatial extent, which we lift into 3D to perform NeRF editing. The strength of this prior is determined by a single parameter, which can be intuitively controlled by a user. By ensuring the mod-

ifications stay within the relevant region, the original input is better preserved (i.e., avoiding extraneous changes), while still satisfying the constraint of the desired edit. Our basic approach is to predict the spatial scope implicit in an input instruction, via the discrepancy between the noise predictions conditioned on the instruction vs. empty text, which we call the *relevance map* (see Figure 1). The edit mask can then be obtained by normalizing and binarizing the relevance map, which we use to preserve unmasked pixels in the denoising process [41].

Our interest is in the editing task of *translation*, where a text-conditioned transformation of an image or 3D scene is performed. In contrast to inpainting, for example, translation is not localized by definition, and hence methods for it tend to over-edit (e.g., see Figs. 4 and 8). Specifically, we focus on isolating the changes incurred in NeRF-based 3D scene translation (i.e., making minimal edits in 3D), by building on the iterative dataset update method of Instruct-NeRF2NeRF (IN2N) [18]. Our approach constructs a multiview-consistent 3D *relevance field* by combining relevance maps across views, enabling localized 3D scene translation. In addition, we show that our approach provides state-of-the-art results for 2D image translation as well.

In summary, (i) we motivate and formulate the new task of localizing edits in 3D, (ii) we present *relevance maps* to predict the spatial scope of an editing instruction on an image, (iii) we use the relevance maps to localize instruction-based image editing in a controllable manner, and (iv) we lift the maps into 3D by *relevance fields* to leverage the localization in scene editing.

2 Related work

Diffusion models for image editing. Diffusion models have shown impressive performance in image synthesis [14, 21, 22, 61, 63, 67, 70]. Text-to-image diffusion models generate high-quality images based on captions [52, 57, 60, 62]. Motivated by this success, pre-trained diffusion models have been used to edit images based on text descriptions [1, 20, 27, 53, 57]. SDEdit [43] adds noise to input images and denoises them conditioned on a desired description, but lacks a mechanism to keep the details of the original image. Recently, IP2P [6] uses Prompt2Prompt [20] to create a dataset, and trains a denoiser conditioned on edit instructions and the original image. IP2P [6] outperforms previous methods, but tends to over-edit images, including parts irrelevant to the instruction. Simply increasing the image guidance scale or reducing the text guidance scale has adverse effects on the region that actually should be edited.

The most similar work to ours in 2D, DiffEdit [11], uses the disagreement between predictions of Stable Diffusion [60] with input and output captions. It thus considers a slightly different problem than IP2P, as it repurposes a more general model, rather than training a new one. Nevertheless, it generally underperforms IP2P, especially for complex cases, and is more difficult to use (requiring two captions, one edited, rather than a single instruction). We therefore build on the state-of-the-art method, IP2P, which does not have a localization mechanism; regardless, however, we provide comparisons to both DiffEdit and IP2P, show-

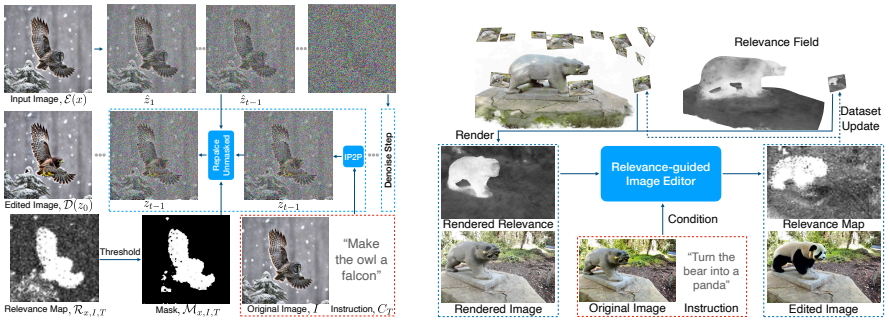


Fig. 2: (a) Overview of a denoising step for image editing via relevance-guidance. The relevance map is binarized to get the edit mask. After each denoising step with IP2P, the unmasked pixels are swapped with the noisy input to ensure consistency to the input throughout the process. (b) Overview of our relevance-guided NeRF editing method. Iteratively, a random view is rendered using both the main NeRF and the relevance field. The rendered relevance then guides the editing of the image render, changing only pixels with highly relevance to the task. IP2P [6], the backbone of the editing method, is always conditioned on the initial scene captures, to prevent drastic drifts from the original scene in the recurrent synthesis process [18]. The relevance-guided editing module (§ 4.2) returns an edited image and an updated relevance, which update the corresponding training views for the NeRF and the relevance field, respectively.

ing superior results in 2D to both in their published forms. Most importantly, DiffEdit does not consider editing 3D scenes, whereas we build on IN2N (an extension of IP2P to NeRFs) to perform localization in 3D.

Editing neural fields. The advent of NeRFs [46] has led to the significant popularity of neural rendering models [72]. NeRFs are getting faster [5, 8, 9, 19, 28, 31, 51, 58, 64, 82], and less data-intensive [24, 34, 35, 39, 55, 77, 79, 83], with improved rendering quality [3, 4, 13, 36, 73]. The popularity of NeRFs naturally introduces a desire for editing tools. Recent works [12, 25, 26, 30, 32, 33, 40, 44, 49, 69, 74, 81, 84] provide NeRF editing approaches, but are typically limited to simple scenes or objects, or perform niche editing tasks, such as color editing. Several works [37, 47, 48, 80] focus on 3D inpainting to remove unwanted objects from NeRFs while yielding a perceptually plausible model. IN2N [18] leveraged IP2P to edit scenes based on text instructions. While IN2N shows promise, it lacks an automated mechanism for localizing edits in 3D, leading to instances of over-editing within scenes. In contrast, our approach includes an automatic localization mechanism for identifying the target region to be edited, thereby restricting modifications to the pertinent area.

In certain cases, manually annotated masks could perform similarly to our automatic localization masks; indeed, several 3D editing methods require user-provided masks or other information (e.g., [41, 47, 48]). However, automated localization has several advantages: (i) our mask is continuously tunable (i.e., a user can control its extent via a threshold); (ii) some prompts may be challenging

for users to create a mask; (iii) it applies to *3D*, where manually providing many view-consistent masks is difficult, and other media (e.g., videos); and (iv) ease of integration into learning pipelines, where manual masks are not scalable.

3 Background

Neural radiance fields. NeRFs [46] represent a 3D scene as a neural field, $f_\theta : (x, d) \rightarrow (c, \sigma)$, mapping a 3D coordinate, $x \in \mathbb{R}^3$, and a view direction, $d \in \mathbb{S}^2$, to a colour, $c \in [0, 1]^3$, and a density, $\sigma \in \mathbb{R}^+$. The field parameters, θ, are optimized to fit the field representation to posed image sets. The field is paired with a rendering operator, implemented as the quadrature approximation of the classical volumetric rendering integral [42]. For a ray, r, parametrized as $r = o+td$, where o is the origin and d is the view-direction, rendering begins with sampling N points, $\{t_i\}_{i=1}^N$, on r between near and far bounds. The rendered colour is then obtained via the volumetric rendering equation, $\widehat{C}(r) = \sum_{i=1}^N w_i c_i$, where $w_i = T_i(1-\exp(-\sigma_i\delta_i))$ is the contribution of the i-th point, $\delta_i = t_{i+1} - t_i$ is the adjacent point distance, and $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j)$ is the transmittance.
InstructPix2Pix. Given an image, I, and a text instruction, C_T, describing the edit, IP2P [6] follows C_T to edit I. IP2P is trained on a dataset where, for each I and C_T, a sample edited image, I_{out}, is given. IP2P is based on Latent Diffusion [60], where a variational autoencoder (VAE) [29] with encoder, $\mathcal{E}$, and decoder, $\mathcal{D}$, is used for improved efficiency and quality. For training, noise, $\epsilon \sim \mathcal{N}(0, 1)$, is added to $z = \mathcal{E}(I_{\text{out}})$ to get the noisy latent, z_t, where the random timestep, $t \in T$, determines the noise level. The denoiser, ϵ_θ, is initialized with pretrained weights [60], and fine-tuned to minimize the diffusion objective, $\mathbb{E}_{I_{\text{out}},I,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, I, C_T))\|_2^2\right]$. During training, conditions are randomly removed [38] by setting $I = \emptyset_I$ and $C_T = \emptyset_T$ to enable unconditional denoising. Thus, the strength of the edit can be controlled by the image guidance scale, s_I, and the text guidance scale, s_T. The modified score estimate is then

$$\widetilde{\epsilon}_\theta(z_t, t, I, C_T) = \epsilon_\theta(z_t, t, \emptyset_I, \emptyset_T) + s_I\Delta_{\epsilon_\theta}(I, \emptyset_I; \emptyset_I, \emptyset_T) + s_T\Delta_{\epsilon_\theta}(I, C_T; I, \emptyset_T), \quad (1)$$

where $\Delta_{\epsilon_\theta}(I_1, C_{T,1}; I_2, C_{T,2}) = \epsilon_\theta(z_t, t, I_1, C_{T,1}) - \epsilon_\theta(z_t, t, I_2, C_{T,2})$. After training, the denoiser can be used to either generate edited images from pure noise, or to iteratively denoise a noisy version of an input image to get an output.

4 Method

To construct our 3D relevance field for localized NeRF editing, we begin by describing the calculation of the 2D relevance map in § 4.1, which we will lift into the multiview setting. In § 4.2, we use the relevance map as a form of mask guidance in 2D. Finally, we integrate this guidance into a neural field to perform localized 3D edits, in § 4.3. Implementation details are in § 4.4.

4.1 Relevance map calculation

Given an image, I, and an edit instruction, C_T, we leverage IP2P [6] to predict the relevance of each pixel to the edit, that is, the likelihood that a given pixel needs to be changed, based on the editing task. First, we add noise to the encoded image, $\mathcal{E}(I)$, until a fixed timestep, t_{rel}, to obtain the noisy latent,

$$z_{t_{\text{rel}}} = \sqrt{\alpha_{t_{\text{rel}}}}\mathcal{E}(I) + \sqrt{1-\alpha_{t_{\text{rel}}}}\,\epsilon, \tag{2}$$

where $\epsilon \sim \mathcal{N}(0,1)$, and α_t is the noise scheduling factor at timestep t. Note that t_{rel} is a constant noise level used in our method as a hyperparameter. We then use IP2P's noise predictor, ϵ_θ, to get two different predictions: (i) the predicted noise conditioned on both the image and the text, $\epsilon_{I,T}(z_{t_{\text{rel}}}) = \epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, C_T)$, and (ii) the predicted noise conditioned only on the image and the empty text as the instruction, $\epsilon_I(z_{t_{\text{rel}}}) = \epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, \text{“”})$. The difference between $\epsilon_{I,T}(z_{t_{\text{rel}}})$ and $\epsilon_I(z_{t_{\text{rel}}})$ is that only the former is aware of the text prompt. We use the magnitude of the mismatch between these two values as a measurement of the relevance of each pixel to the edit. To this end, we first calculate the absolute difference between the two values, which we call the *relevance map*,

$$\mathcal{R}_{x,I,T} = |\epsilon_{I,T}(z_{t_{\text{rel}}}) - \epsilon_I(z_{t_{\text{rel}}})|. \tag{3}$$

For robustness, we further clamp the outlier values using the interquartile range (IQR) with ratio 1.5, and normalize the relevance map between 0 and 1. Figure 1 (left inset) illustrates an overview of the calculation of the relevance mask. For 3D editing, our approach obtains relevance maps of each image in the multiview setting, and consolidates them into a separate neural relevance field.

4.2 Relevance-guided image editing

The relevance map guides the generation of the edited image, by localizing the edited region. For a pixel, a high relevance value means that the pixel is likely to be relevant to the edit, and we allow it to change. In contrast, a low relevance map value signals that the pixel is unlikely to require change. We apply a single mask threshold, $\tau \in [0,1]$, on the relevance map to obtain the edit mask, $\mathcal{M}_{x,I,T} = \mathbb{1}(\mathcal{R}_{x,I,T} \geq \tau)$, enclosing the pixels we allow to be edited. To edit an input image, x, its encoding, $\mathcal{E}(x)$, is diffused to a fixed noise level, t_{edit}, to get the starting noisy latent, $z_{t_{\text{edit}}}$. The edit noise level, t_{edit}, determines the strength of the edit; setting it to 0 results in the input image being unchanged, while setting it to the maximum diffusion timestep starts the generation from pure noise. Each denoising stage takes a noisy latent, z_t, and denoises it to get z_{t-1}. The denoising step begins with predicting the noise via IP2P to get $\widetilde{\epsilon}_t = \widetilde{\epsilon}_\theta(z_t, t, I, C_T)$. Using $\widetilde{\epsilon}_t$ and DDIM [68], the mask-unaware prediction at timestep $t-1$ is

$$\widetilde{z}_{t-1} = \sqrt{\alpha_{t-1}}\Big(\frac{x_t - \sqrt{1-\alpha_t}\,\widetilde{\epsilon}_t}{\sqrt{\alpha_t}}\Big) + \sqrt{1-\alpha_{t-1}}\,\widetilde{\epsilon}_t. \tag{4}$$

The unedited noisy latent of the input image, x, at timestep $t-1$ would have been $\widehat{z}_{t-1} = \sqrt{\alpha_{t-1}}\mathcal{E}(x) + \sqrt{1-\alpha_{t-1}}\epsilon$. To obtain z_{t-1}, we combine the mask unaware prediction, $\widetilde{z}_{t-1}$, and the unedited noisy latent, $\widehat{z}_{t-1}$, as

$$z_{t-1} = \widetilde{z}_{t-1} \odot \mathcal{M}_{x,I,T} + \widehat{z}_{t-1} \odot (1 - \mathcal{M}_{x,I,T}). \tag{5}$$

This way, by replacing the unmasked pixels with the noisy version of the input image, we constrain the generation process from changing any pixel outside of the mask. After iterative denoising, the edited image, $\mathcal{D}(z_0)$, is obtained. Figure 2a presents an overview of our denoising process.

Interpretation. The relevance map is closely related to Score Distillation Sampling (SDS) [55, 76], a method of flowing gradients to a model through a mapping to images (e.g., rendering, for 3D generation). In particular, following prior work [78], assume we wish to generate an edited image, O, that matches the diffusion model prior. Then, $\mathfrak{L}(O) = \mathbb{E}_\epsilon\big[\log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I) - \log p_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I, C_T)\big]$, approximates our objective, where q^O is the forward diffusion process, p is the reverse denoising process, and z_t is the encoded diffused latent at time t, which depends on the noise, ϵ. Using ϵ_θ, among other assumptions [55, 59], we have

$$\nabla_O \mathfrak{L} \approx \frac{\alpha_{t_{\text{rel}}}}{\sigma_{t_{\text{rel}}}} \mathbb{E}_\epsilon\big[\underbrace{\epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, C_T) - \epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, \emptyset_T)}_{\text{signed relevance map, } \widetilde{\mathcal{R}}_{x,I,T}}\big] \frac{\partial \mathcal{E}(O)}{\partial O}, \tag{6}$$

where (by Eq. 3) we have $\mathcal{R}_{x,I,T} = |\widetilde{\mathcal{R}}_{x,I,T}|$. Intuitively, since the SDS gradient (Eq. 6) approximates an update that follows the model score [55], *the relevance map weights how much each pixel should be altered* by such an update. Our post-processing (e.g., into $\mathcal{M}_{x,I,T}$) amplifies this natural weighting. Hence, we can interpret it as a controllable way to emphasize the essential parts of the edit, while suppressing spurious changes. Please see supplement for details (including proofs). To our knowledge, this connection has not been previously made.

There is also a relation to classifier-free guidance (CFG) [23], which linearly combines conditional and unconditional score estimates (see Eq. 1). In contrast, our approach takes the difference between those scores, non-linearly normalizes it, and then uses the resulting mask to spatially localize the update (Eq. 5).

4.3 Relevance field for 3D scene editing

Our primary application is the localization of edits in 3D scenes, extending the approach applied in 2D. Given a multiview capture, $\{I_i\}_{i=1}^n$, of a static scene and the corresponding camera poses, the goal is to edit a NeRF fit to the scene, f_θ, according to a text prompt, C_T. Motivated by IN2N [18], we perform iterative training view updates by replacing one training view, I_t, at a time by its edited counterpart, based on C_T. To ensure consistency in the edit localizations across views, we fit a 3D neural field, which we call the *relevance field*, to the relevance maps of all the training views. While editing each of the views, we render the corresponding relevance map from the relevance field to guide the edit.

For implementation, we extend the neral field function, f_θ, to return a view-*in*dependent relevance, $r(x) \in [0, 1]$, for every 3D coordinate, $x \in \mathbb{R}^3$. Notice that the geometry of the main NeRF and the relevance field is shared, and when fitting the relevance field, we always detach the gradients of the densities to ensure that the potential inconsistencies do not affect the geometry of the main scene. For a ray, r, the rendered relevance, $\widehat{R}(r)$, can be simply calculated by replacing the point-wise colours with relevance values in the volumetric rendering equation, via $\widehat{R}(r) = \sum_{i=1}^{N} w_i r_i$.

During the NeRF editing process, every n_{edit} iterations, we randomly sample a training view, I_i. The first time we sample I_i, the relevance map, $\mathcal{R}_{I_i,I,T}$, is calculated and added to the training data of the relevance field. From the same view, the image, $\widehat{I}_i$, and the relevance map, $\widehat{R}_i$, are rendered using f_θ. The relevance-guided image editor from § 4.2 is used to locally edit $\widehat{I}_i$, conditioned on the original captured image, I_i, and the text condition, C_T. To this end, the encoded render, $\mathcal{E}(\widehat{I}_i)$, is diffused until a random timestep, $t_{\text{edit}} \in T$, to obtain $z_{t_{\text{edit}}}$. The noisy latent, $z_{t_{\text{edit}}}$, is iteratively denoised conditioned on the unedited view, I_i, and the text, C_T, guided by the rendered relevance map, $\widehat{R}_i$, to obtain the edited training view, $\widetilde{I}_i = \mathcal{D}(z_0)$. Since the several-fold upsampling induced by the decoder could lead to inconsistencies in the unedited region, we replace the unedited RGB pixels in $\widetilde{I}_i$ with their counterparts from I_i using a relevance mask rendered in the original image resolution. After editing, $\widetilde{I}_i$ replaces the corresponding training view to supervise the main NeRF (the colour field).

4.4 Implementation details

In all of our experiments, we set $t_{\text{rel}} = 0.8$, i.e., we apply 80% of the noise to predict the relevance map. For IP2P, we use the model available on HuggingFace, based on the diffusers package [54]. For NeRF editing, we use the nerfacto model from NeRFStudio [71]. During the iterative dataset updates, we perform edits with noise levels (timesteps) randomly sampled from $[0.02, 0.98]$. We update a single training view every $n_{\text{edit}} = 10$ iterations. Each image is updated using 20 denoising steps for NeRF editing, and 100 denoising steps for image editing. For the relevance field implementation, we use the same hyperparameters as the nerfacto field [71]; however, we never use the densities from this field, and only use the geometry from the main radiance field. The threshold, τ is set between $[0.4, 0.6]$ in all the experiments, unless stated otherwise. Each NeRF is first trained for 30,000 iterations on the original scene, and then edited for 3,000 or 4,000 iterations depending on the number of training views.

5 Experiments

While our focus is on automatic localization of 3D edits in the context of NeRF scene translation, where we improve upon the unlocalized IN2N, our approach in 2D is also state of the art, specifically building upon IP2P and DiffEdit.

Datasets. NeRF editing evaluation is done using scenes from IN2N [18] and

LLFF [45]. We use 14 different NeRF editing tasks (i.e., text instructions) for the quantitative experiments. For each, a scene is edited using an instruction, and evaluated against a desired output caption. IN2N and LLFF provide multiview captures of forward-facing and 360° static scenes. Colmap [65, 66] is used to recover camera parameters. For image editing, we follow IP2P [6] and use their dataset of diverse images and editing instructions. The test set consists of 5,000 images, paired with instructions, and input and output captions.

Metrics. For scene editing, we use CLIP text-image similarity (Txt-Img Sim.) which is the cosine similarity of the CLIP embeddings of output views and the output caption. In addition, CLIP frame similarity (Frame Sim.) measures the cosine similarity of consecutive frames. CLIP edit similarity (Edit Sim.) measures the directional agreement between the changes applied to neighbouring frames in CLIP space, as described in IN2N [18]. CLIP image similarity (Image Sim.) and edit PSNR measure the consistency of the edited views and the input views, in the CLIP and RGB spaces, respectively. Finally, we use NIQE [50], a no-reference image quality metric, to evaluate the quality and sharpness of the outputs. In 2D, following IP2P [6], we use CLIP image similarity [56] (i.e., Image Sim. in 3D) and CLIP text-image direction similarity [16].

Baselines. In 3D, we evaluate against IN2N [18] and *per-frame* IP2P (IP2P-PF), which independently edits renders of the input NeRF via IP2P [6]. Note that using IP2P in this manner is an *upper-bound* on image quality (as it removes NeRF artifacts), but should have worse multiview consistency (as it has no 3D structure). We further compare our model against NeRF-Art [75], which uses CLIP similarity of the scene and a caption to edit scenes, IN2N [18] with stable diffusion (SD) [60] (rather than IP2P), and using the Score Distillation Sampling (SDS) [55] loss with IP2P. For 2D image editing, we compare against DiffEdit [11], SDEdit [43], and IP2P [6]. DiffEdit expects input and output captions, and is evaluated with those, instead of the edit instruction. SDEdit expects the output caption; we evaluate it with the output caption as *SDEdit (out caption)* and with the edit instruction as *SDEdit (instruction)*, separately.

5.1 Results

NeRF editing. Table 1 contains quantitative results based on 14 scene editing tasks (i.e., text instructions). All methods perform comparably in terms of CLIP [56] text-image similarity, meaning the edited scenes match the output captions. Since IP2P-PF *directly outputs from a diffusion model*, rather than a NeRF, it (i) has no 3D-awareness and (ii) bounds image quality performance (i.e., since it lacks NeRF artifacts, the other methods will have worse NIQE). Nevertheless, CLIP frame similarity and CLIP edit similarity show that IN2N and our method produce view-consistent results, whereas IP2P-PF independently edits rendered views and is unable to maintain 3D consistency. However, since IN2N tends to over-edit, our method is superior in CLIP image similarity and Edit PSNR, which measure consistency with the original scene. Further, in terms of NIQE [50], our method outperforms IN2N by producing sharper and higher quality results. In summary, our method maintains the 3D view consistency and

Table 1: 3D scene editing results, compared to IN2N [18] and *per-frame* IP2P [6] (IP2P-PF). Since IP2P-PF *independently* edits each image, (i) there are no 3D consistency constraints and (ii) NeRF artifacts are not present (thus *bounding* image quality performance). Hence, it is not a direct comparison. Nevertheless, due to (i), it is inferior to the other methods in terms of consistency between neighbouring frames and their edited forms. Compared to IN2N, our approach essentially matches its Txt-Img, Frame, and Edit Similarities; however, we surpass it in (a) similarity to the original scene (Image Sim. and Edit PSNR) and (b) overall image quality (NIQE).

Method	Txt-Img Sim.↑	Frame Sim.↑	Edit Sim.↑	Image Sim.↑	Edit PSNR↑	NIQE↓
IP2P-PF [6]	0.28	0.97	0.81	0.91	19.4	4.02
IN2N [18]	0.27	0.99	0.88	0.86	28.7	6.43
Ours	0.27	0.99	0.88	**0.89**	**31.0**	**5.53**

Fig. 3: Qualitative outputs of our NeRF editing method. For each scene and text instruction, we provide multiview renderings of the edited NeRF to show view consistency. Our method follows the text, yet keeps regions less relevant to the task intact.

semantic similarity to the desired edit of IN2N, but improves both similarity to the original scene and overall image quality. Quantitatively, we find that *our approach closes 37.3% of the gap in NIQE between IN2N and IP2P-PF.*

Qualitative examples of our scene editing results are shown in Figure 3. Our model edits the region most relevant to the edit, while keeping the rest of the scene unchanged. For example, while editing the bear statue and changing it to a *panda*, a *grizzly bear*, or a *polar bear*, the background and the stage underneath the statue remain intact, while the statue itself is changed to the desired animals with sharp textures (notice the texture of the fur).

We next show qualitative comparisons to the baselines in Figure 4. We see that IN2N has a tendency to over-edit scenes, as it is built directly on IP2P. For instance, in the leftmost inset, it has changed the entire torso to a bronze statue, not just the face. In the "Give him blue hair" inset, notice how it has also changed the T-shirt, eyes, and background colours. In the bear scene, the background in IN2N outputs is blurred. This is due in part to the ambiguity of the VAE decoder in upsampling, resulting in minor misalignments between different views.

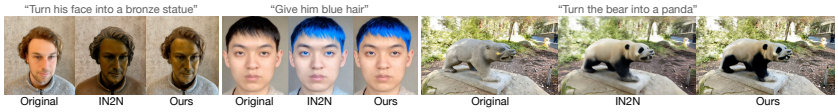
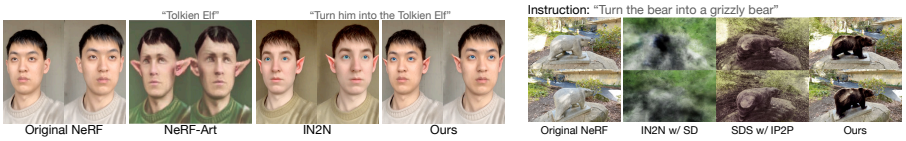


Fig. 4: Comparison of our 3D scene editing results against IN2N [18]. The relevance field enables us to localize the edit to the most significant regions. Editing a smaller region reduces the decoder spatial ambiguity problem on unedited pixels. Moreover, it improves the view consistency in the edited region as editing a small part is more likely to produce consistent results across the views.



(a) **(b)**

Fig. 5: (a) Comparison of our 3D scene editing method against NeRF-Art [75] and IN2N [18]. The baselines modify the background, shirt, and hair of the person, while our model only edits the eyes and ears. The extraneous changes of the baselines can even fail to preserve important scene semantics (in this case, the individual's identity). In contrast, our method applies only the minimum change required for the desired semantic alteration. (b) Qualitative comparison of our scene editing method against two baselines. *IN2N w/ SD* performs the same iterative dataset updates as IN2N [18], but with stable diffusion [60] instead of IP2P. *SDS w/ IP2P* performs updates on the NeRF based on the SDS loss [55] calculated via IP2P. Our method results in sharp outputs, while the baselines have failed on the task.

Moreover, since IP2P fails to prevent changes to the background, some of the edited views have an altered background. This inconsistency reduces sharpness. It also disrupts the optimization, as network capacity and loss gradients are allocated to background inconsistencies; hence, IN2N outputs are not as sharp as our result. Moreover, IP2P is constrained to only edit the bear to a panda in our case, rather than trying to edit the entire image to satisfy the instruction. Consequently, the edited views in our method are more likely to be consistent, especially for nearby views, which is another reason that even our *edited* regions are considerably sharper (e.g., the texture of the panda's fur).

In Figure 5a, NeRF-Art [75] follows the instruction and changes the face to the *Tolkien Elf*, but the edited scene has quality artifacts associated with CLIP-based [56] methods, and has changed irrelevant regions of the scene, including the hair, background, and t-shirt. Figure 5b compares our method with additional baselines. The baselines either have a high failure rate [11] or result in global image updates and have a lack of detail preservation [43].

Image editing. While our main contribution is the 3D relevance field, our method is also state of the art for text-guided image translation. We quantitatively evaluate our relevance-guided image editing method in Figure 6, based on the IP2P [6] dataset. Each model is measured on two competing metrics: similar-

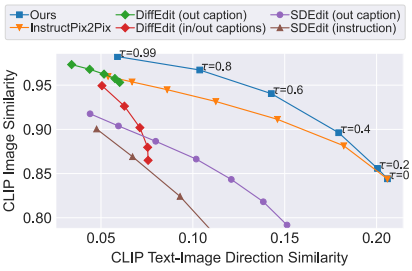


Fig. 6: Quantitative 2D image editing evaluation. Our model achieves better text-image direction similarity (x-axis), while maintaining higher fidelity to the input (y-axis). We set $s_T = 7.5$ for every method. We pick SDEdit's strength from $[0.1, 0.9]$ and DiffEdit's encoding-ratio from $[0.5, 0.9]$. For IP2P, $s_I \in [1, 2.2]$. For our method, $s_I = 1$. Notice that our method is best for both metrics *for every choice of* τ.

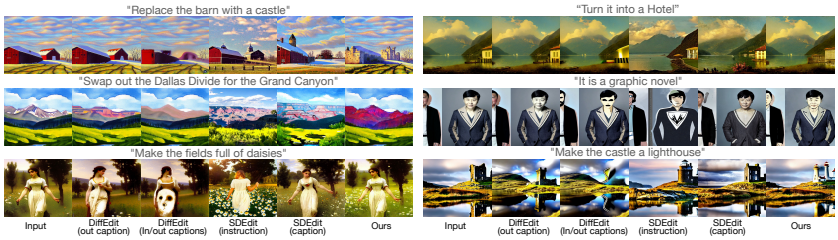


Fig. 7: Comparison of our image editing method against DiffEdit [11] and SDEdit [43]. DiffEdit requires the captions of both the input and output, but still fails to perform the edit as the captions in IP2P [6] dataset are relatively complex. SDEdit [43] performs better when it is given the output caption. Our model follows the instructions more closely, while maintaining coherence with the input. See also Figure 8.

ity to the input image (y-axis) and agreement with the edit (x-axis). Compared to the baselines, our model achieves higher image consistency with similar directional similarities. Additionally, increasing the mask threshold, τ, increases the image similarity as a smaller image region is being edited. However, overly increasing τ can restrict the edit too much. Nevertheless, for these metrics, our method is on the Pareto frontier *for every value of* τ.

We provide qualitative comparisons in Figures 7 and 8. Compared to our closest competitor, IP2P, we consistently obtain superior image similarity, by avoiding over-editing. In terms of localization, the most similar method to ours, DiffEdit [11], requires access to both input and output captions. Even with this information, since the captions in the IP2P dataset are relatively complex (rather than simple class names or high-level descriptions), DiffEdit fails to perform appropriate edits. In particular, when DiffEdit is only given the output caption and an empty text as the input caption, i.e., *DiffEdit (out caption)*, it never achieves high text-image similarities, and the inputs remain relatively unchanged. This is due in part to DiffEdit failing to predict appropriate masks. For SDEdit [43,60], the fidelity of the outputs to the inputs drop significantly as the strength of the edit is increased. This drop is due to the lack of an explicit mechanism to ensure consistency. Unlike our model, SDEdit relies on the information in the noisy latent, however, in later diffusion stages, the noisy latent retains global information about the input, but lacks local details. Overall, these results showcase the

Fig. 8: Comparison of our image editing method against IP2P [6]. For both models, $s_T = 7.5$ and $s_I = 1$. IP2P fails to isolate the specified region, and over-edits the input. Our model explicitly predicts the scope of the edit, and limits it to a specific region.

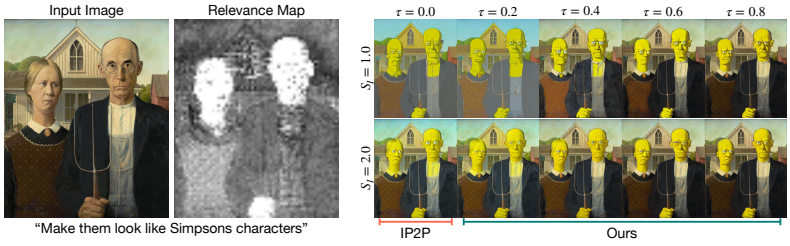


Fig. 9: Comparison of the mask threshold, τ, and the image guidance scale, s_I. Increasing s_I improves the similarity of the output and the input image, but significantly decreases the intensity of the edit overall. In contrast, τ provides a way for the user to control the *region* of the edit, without changing the *strength* of the edit.

utility of *minimal edits*: by localizing edits to only the areas relevant to meeting the desired semantics, our method produces text-image similarities on-par with IP2P, while keeping the outputs more consistent with the inputs.

Mask Threshold. We explore how the mask threshold, τ, affects the editing process (Figure 9). Setting $\tau = 0$ results in every pixel being masked, which is equivalent to IP2P [6]. For each τ, we provide results with different image guidance scales, s_I. As evident in the results, increasing s_I is insufficient to localize the IP2P edits; instead, it merely weakens the overall edit itself (reducing text-image similarity in Figure 6). On the other hand, changing τ provides a different form of control to the user, and allows them to control the edited region, without negatively impacting regions that do not need modification.

Relevance noise level. The relevance noise level, t_{rel}, controls the diffusion time used to derive our relevance map and field. Figure 10 compares maps calculated using different levels. Empirically, we found $t_{\text{rel}} = 0.8$ to be reliable. This way, the relevance is calculated using predictions in the higher-noise stages. As a result, the noise estimator fixates on the global structure of the generated images, rather than the fine details [2]. Thus, the predicted relevance masks encapsulate the global boundaries of the relevant regions. Moreover, Figure 10 shows the

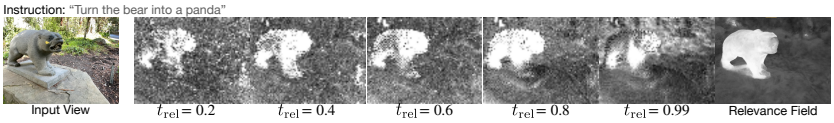


Fig. 10: Relevance maps across t_{rel} values vs. a render from the relevance field.

Fig. 11: Example failure cases. Although our model outperforms IP2P, it still relies on it for predicting the relevance. Hence, it cannot recover from IP2P's errors.

render of the relevance *field* from the same view. Since the field is supervised using maps from multiple views, it acts as an ensemble over relevance predictions, which is more accurate than each single map. In addition, the inductive bias of the NeRF architecture limits high-frequency variations; hence, relevance renders provide a smooth consensus over the 3D scene, with minimal noise.

Failure cases. Although our mask-guidance can alleviate the over-editing problem of IP2P [6], while reducing upsampling ambiguity, it is still unable to recover from cases where IP2P fails. In Figure 11, we provide examples of such failures. For instance, in the first row, the prompt is *"change to a rosé"*. Given the image context, the goal is to only change the drinks. However, IP2P has changed the background field and hair colour to pink. This failure is reflected in the predicted relevance mask, which superfluously highlights those areas. Though our model reduces these over-edits, it has still produced unnecessary changes. In the second example, "*add a cat*", localizing the edit with respect to the prompt is an ambiguous problem. The relevance map has failed to localize a certain position for the cat to be added, and instead, the person and the dog have been replaced with cats. Our method is agnostic to the underlying instruction-conditioned diffusion model, and can benefit from swapping IP2P with a better one in the future.

6 Conclusion

We presented a method for predicting the relevance of each image pixel to an editing task based on a text instruction. The relevance map is defined as the discrepancy between a conditional and an unconditional pass over a diffusion-based image editor. The relevance is used as a mask to guide the image generation process and force the unmasked pixels to not change, resulting in a localized image editor. We further showed that training a relevance field on the relevance maps of the training views of a NeRF achieves similar localizations when editing 3D scenes. Empirically, our method has superior performance compared to the baselines in both 3D scene editing and (single) image tasks.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: CVPR (2022) 3
2. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., Liu, M.Y.: ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. arXiv (2022) 13
3. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. ICCV (2021) 4
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. CVPR (2022) 4
5. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-NeRF: Anti-aliased grid-based neural radiance fields. ICCV (2023) 4
6. Brooks, T., Holynski, A., Efros, A.A.: InstructPix2Pix: Learning to follow image editing instructions. In: CVPR (2023) 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 19, 20, 22, 23
7. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2Video: Video editing using image diffusion. In: ICCV (2023) 1
8. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022) 4
9. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: MobileNeRF: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: CVPR (2023) 1, 4
10. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint arXiv:2305.06558 (2023) 25, 27
11. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. ICLR (2023) 1, 2, 3, 9, 11, 12
12. Dai, A., Siddiqui, Y., Thies, J., Valentin, J., Niessner, M.: SPSG: Self-supervised photometric scene generation from RGB-D scans. In: CVPR (2021) 4
13. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised NeRF: Fewer views and faster training for free. In: CVPR (2022) 4
14. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. NeurIPS (2021) 3
15. Dong, J., Wang, Y.X.: Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. In: NeurIPS (2023) 1
16. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: StyleGAN-NADA: Clip-guided domain adaptation of image generators. TOG (2022) 9
17. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: TokenFlow: Consistent diffusion features for consistent video editing. In: ICLR (2024) 1
18. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-NeRF2NeRF: Editing 3D scenes with instructions. ICCV (2023) 1, 2, 3, 4, 7, 8, 9, 10, 11
19. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: ICCV (2021) 4
20. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. ICLR (2023) 1, 3
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020) 3

22. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. JMLR (2022) 3
23. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 7
24. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. CVPR (2022) 4
25. Jheng, R.F., Wu, T.H., Yeh, J.F., Hsu, W.H.: Free-form 3D scene inpainting with dual-stream GAN. BMVC (2022) 4
26. Kania, K., Yi, K.M., Kowalski, M., Trzciński, T., Tagliasacchi, A.: CoNeRF: Controllable neural radiance fields. In: CVPR (2022) 4
27. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. CVPR (2023) 3
28. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian splatting for real-time radiance field rendering. TOG (2023) 4
29. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv (2022) 5
30. Kuang, Z., Luan, F., Bi, S., Shu, Z., Wetzstein, G., Sunkavalli, K.: PaletteNeRF: Palette-based appearance editing of neural radiance fields. In: arXiv (2022) 4
31. Kurz, A., Neff, T., Lv, Z., Zollhöfer, M., Steinberger, M.: AdaNeRF: Adaptive sampling for real-time rendering of neural radiance fields. In: ECCV (2022) 4
32. Lazova, V., Guzov, V., Olszewski, K., Tulyakov, S., Pons-Moll, G.: Control-NeRF: Editable feature volumes for scene rendering and manipulation. In: WACV (2023) 4
33. Li, Z., Fan, T., Li, Z., Cui, Z., Sato, Y., Pollefeys, M., Oswald, M.R.: CompNVS: Novel view synthesis with scene completion. In: ECCV (2022) 4
34. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3D: High-resolution text-to-3D content creation. In: CVPR (2023) 4
35. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: BARF: Bundle-adjusting neural radiance fields. In: ICCV (2021) 4
36. Lindell, D.B., Van Veen, D., Park, J.J., Wetzstein, G.: BACON: Band-limited coordinate networks for multiscale scene representation. In: CVPR (2022) 4
37. Liu, H.K., Shen, I.C., Chen, B.Y.: NeRF-In: Free-form NeRF inpainting with RGB-D priors. In: arXiv (2022) 4
38. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models. ECCV (2023) 5
39. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3D object. arXiv (2023) 4
40. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. In: ICCV (2021) 4
41. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: CVPR (2022) 1, 2, 3, 4
42. Max, N., Chen, M.: Local and global illumination in the volume rendering integral. Tech. rep., Lawrence Livermore National Lab (LLNL), Livermore, CA (United States) (2005) 5
43. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. ICLR (2021) 1, 2, 3, 9, 11, 12
44. Mikaeili, A., Perel, O., Safaee, M., Cohen-Or, D., Mahdavi-Amiri, A.: SKED: Sketch-guided text-based 3D editing. arXiv (2023) 4

45. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. TOG (2019) 9
46. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1, 4, 5
47. Mirzaei, A., Aumentado-Armstrong, T., Brubaker, M.A., Kelly, J., Levinshtein, A., Derpanis, K.G., Gilitschenski, I.: Reference-guided controllable inpainting of neural radiance fields. In: ICCV (2023) 1, 4
48. Mirzaei, A., Aumentado-Armstrong, T., Derpanis, K.G., Kelly, J., Brubaker, M.A., Gilitschenski, I., Levinshtein, A.: SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In: CVPR (2023) 1, 4
49. Mirzaei, A., Kant, Y., Kelly, J., Gilitschenski, I.: LaTeRF: Label and text driven object radiance fields. In: ECCV (2022) 4
50. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal Processing Letters (2013) 9
51. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. TOG (2022) 1, 4
52. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In: ICML (2022) 3
53. Parmar, G., Singh, K.K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. In: SIGGRAPH (2023) 3
54. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers (2022) 8
55. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFusion: Text-to-3D using 2D diffusion. In: ICLR (2023) 4, 7, 9, 11, 22, 23, 24
56. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. ICML (2021) 9, 11
57. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv (2022) 3
58. Reiser, C., Szeliski, R., Verbin, D., Srinivasan, P.P., Mildenhall, B., Geiger, A., Barron, J.T., Hedman, P.: MERF: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. In: arXiv (2023) 1, 4
59. Roeder, G., Wu, Y., Duvenaud, D.: Sticking the landing: Simple, lower-variance gradient estimators for variational inference. arXiv (2017) 7, 23, 24
60. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 3, 5, 9, 11, 12
61. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: SIGGRAPH (2022) 3
62. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022) 3
63. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. TPAMI (2022) 3
64. Sara Fridovich-Keil and Alex Yu, Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: CVPR (2022) 4

65. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 9
66. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016) 9
67. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. PMLR (2015) 3
68. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. ICLR (2021) 2, 6
69. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: CVPR (2017) 4
70. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. NeurIPS (2020) 3
71. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A modular framework for neural radiance field development. In: SIGGRAPH (2023) 8
72. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Wang, Y., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., Simon, T., Theobalt, C., Niessner, M., Barron, J.T., Wetzstein, G., Zollhoefer, M., Golyanik, V.: Advances in neural rendering. In: SIGGRAPH (2021) 4
73. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-NeRF: Structured view-dependent appearance for neural radiance fields. CVPR (2022) 4
74. Wang, C., Chai, M., He, M., Chen, D., Liao, J.: CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. CVPR (2022) 4
75. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: NeRF-Art: Text-driven neural radiance fields stylization. TVCG (2023) 9, 11
76. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score Jacobian chaining: Lifting pretrained 2D diffusion models for 3D generation. In: CVPR (2023) 7, 24
77. Wang, Q., Wang, Z., Genova, K., Srinivasan, P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: IBRNet: Learning multi-view image-based rendering. In: CVPR (2021) 4
78. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. arXiv (2023) 7, 23
79. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: NeRF--: Neural radiance fields without known camera parameters. In: arXiv (2021) 4
80. Weder, S., Garcia-Hernando, G., Monszpart, A., Pollefeys, M., Brostow, G., Firman, M., Vicente, S.: Removing objects from neural radiance fields. In: CVPR (2023) 4
81. Yang, B., Zhang, Y., Xu, Y., Li, Y., Zhou, H., Bao, H., Zhang, G., Cui, Z.: Learning object-compositional neural radiance field for editable scene rendering. In: ICCV (2021) 4
82. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: PlenOctrees for real-time rendering of neural radiance fields. In: ICCV (2021) 4
83. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. In: CVPR (2021) 4
84. Yuan, Y.J., Sun, Y.T., Lai, Y.K., Ma, Y., Jia, R., Gao, L.: NeRF-editing: geometry editing of neural radiance fields. In: CVPR (2022) 4
85. Zhang, Z., Li, B., Nie, X., Han, C., Guo, T., Liu, L.: Towards consistent video editing with text-to-image diffusion models. NeurIPS (2024) 1