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

Supplementary Material

A Mask threshold effect

Figure 12 provides a quantitative comparison of our image editing method with different mask thresholds against IP2P [6]. Our method produces outputs that closely reflect the desired edits (x-axis), while remaining consistent with the inputs (y-axis) by confining the edits in the relevant region. However, while increasing the mask threshold results in higher fidelity to the input, overly increasing it can prevent the model from editing the parts that actually matter: the lines in Figure 12 cover a smaller text-image direction similarity region as the mask threshold, τ, is increased. Based on this experiment, we set τ within $[0.4, 0.5]$ throughout the paper.

We further provide a qualitative example to showcase the effect of the mask threshold on the generation of the edited image (Figure 13). Setting the mask threshold, τ, to 0 results in every pixel being masked. As a result, our model with $\tau = 0$ is equivalent to IP2P [6]. For each τ, we provide results with different image guidance scales, S_I. As evident in the results, in IP2P, simply increasing the image guidance scale is not enough to localize the edits; with $S_I = 1.0$, the background and the clothes are drastically changed. When setting $S_I = 3.0$ in IP2P, the woman's collar and the man's shirt are still changed to yellow, while the faces no longer look like Simpsons characters; increasing S_I has an adverse effect on the text-image similarity, which is consistent with our quantitative findings in Figures 6 (main paper) and 12. Alternatively, changing τ provides a different guidance knob to the user, and allows them to control the region to be edited, with minimal damage to the regions that actually need to be modified.

B Additional qualitative comparisons

In Figure 14, we provide additional examples to compare our localized image editing method with IP2P. In all of these examples, S_I is between $[0.8, 1.0]$, and S_T is always 7.5. Mask thresholds are either 0.4 or 0.5. For each of the examples, we further provide the relevance map predicted by our method. Our results are more consistent with the input image by only locally changing the inputs in the regions with high relevance values. Meanwhile, our method follows the instructions closely, and yields images with similar or better edit qualities compared to IP2P.

C Sample relevance field renderings

In Figure 15, we visualize sample relevance fields trained on different scenes and different edit instructions. Each relevance field is trained via 2D supervision from

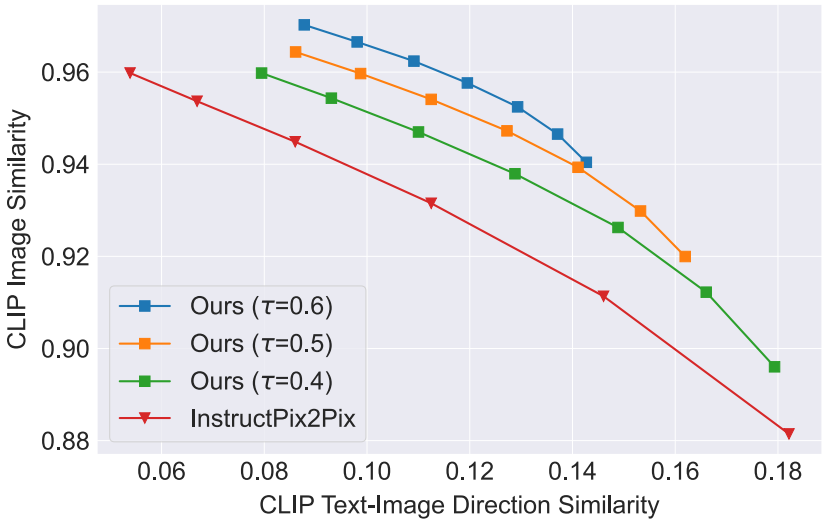

Fig. 12: Quantitative comparison of our image editing method with different mask threshold values, τ, against IP2P [6]. The text-guidance is set to 7.5 for both methods. For IP2P and our method, S_I is changed between $[1.2, 2.2]$ and $[1.0, 2.2]$, respectively.

relevance maps of the training views. As shown in the results, the relevance fields are mainly activated around the intended region to be edited. To edit a NeRF, we use the rendered views from the relevance field as relevance-guidance for editing training views. As a result, the updates of training views during the iterative update process are only locally changed, and the changed region is consistent across different views.

Note that the density of the relevance field (i.e., its geometry) is always queried from the main NeRF model that is being edited. This is to ensure that potential inconsistencies between 2D relevance maps of different views do not impair the main NeRF, and to enforce 2D relevance maps to be projected to the actual geometry of the scene and thus become 3D-consistent; otherwise, the relevance field's geometry might converge to a degenerate solution to justify the inconsistencies. As the main NeRF is being updated towards the desired edited NeRF, its geometry might change. In that case, since the relevance field shares the same geometry, its 2D maps will be projected to the updated NeRF. Thus, during each update, the relevance field localizes the edits on the current version of the main NeRF. This allows slow changes in the geometry of the scene when necessary, while only locally updating the views at each step (e.g., consider the addition of the mustache or the sunglasses, which require changes to the densities).

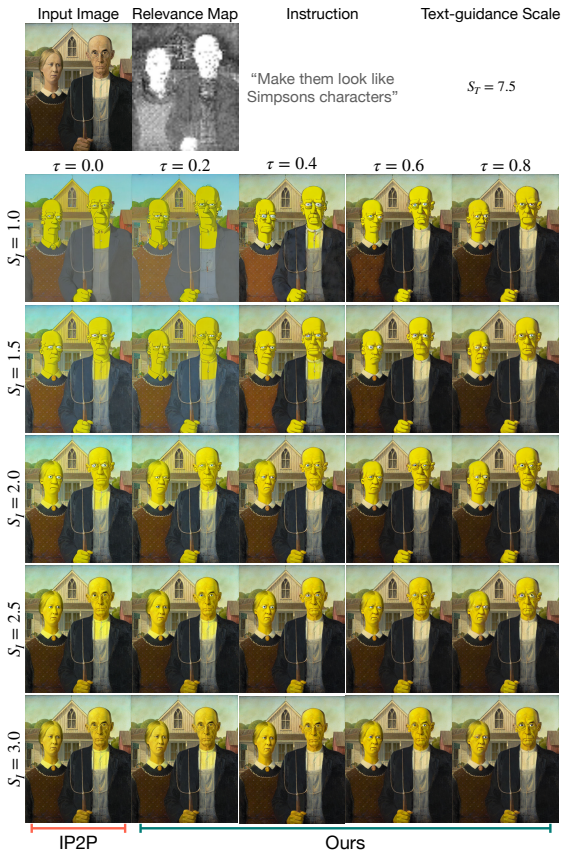

Fig. 13: Qualitative comparison of the effects of the mask threshold, τ, and the image guidance scale, S_I. Increasing the image guidance scale improves the similarity of the output and the input image, but significantly decreases the intensity of the edit, resulting in a failed edit. In contrast, τ provides a knob to the user to control the *region* of the edit, without changing the *strength* of the edit.

D Additional Details

For Figure 8 (main paper), we set S_T and S_I to 7.5 and 1, respectively, while selecting a suitable τ to each edit. In Figure 7 (main paper), for our method, we set $S_T = 7.5$, $S_I = 0.8$, and $\tau = 0.4$. For the other methods we use their default set of hyperparameters. In Figures 11 and 14 (both in the main paper), we set $S_T = 7.5$, $S_I = 0.8$ for both methods and $\tau = 0.4$ for our method. For NeRF editing experiments, τ is always set to 0.5, and the guidance scales are as follows:

- **Bear**: $S_T = 6.5, S_I = 1.5$
- **Face**: $S_T = 7.5, S_I = 1.5$

Fig. 14: Additional qualitative comparison of our image editing results against IP2P [6]. For each image, we show the instruction, outputs of IP2P and our model, and the relevance map predicted by our model. Our model follows the instruction, while maintaining more consistency with the input image. This is due to the relevance-guidance, as only pixels with high relevance values can be modified.

- **Farm**: $S_T = 12.5, S_I = 1.5$
- **Fangzhou**: $S_T = 6.5, S_I = 1.5$

E Expanded interpretation of the relevance map

In this section, we begin by modeling the image editing task as an optimization problem, closely related to score distillation sampling (SDS) [55]. We then show that the relevance map naturally appears in the formulation of updating an initial image towards the edited one based on an SDS-like [55] update scheme. In particular, the relevance map (in its unnormalized signed form) appears as a coefficient in the gradient of a consistency loss, which demands that the distribution induced by the forward diffusion process matches that of the reverse one, in the text-conditioned context. Such a loss can be interpreted in the sense

of variational inference (e.g., [59]), and is closely related to techniques designed to generate 3D textured shapes from 2D diffusion models (e.g., [55, 78]). Intuitively, SDS-based approaches perturb an image with noise and then move along the score function to a higher density area. In contrast, as we show next, in this theoretical SDS-like optimization, the relevance map instead follows a modified score estimate, which effectively subtracts out the component of the score direction that is not "edit-specific" (i.e., based on the input text instruction). In practice, this difference highlights areas of high vs. low importance to the semantically aware editor.

Suppose we want to edit an image, I, with a text instruction, C_T, to generate an output image, O. We denote the optimal edited image as O^*. Motivated by DreamFusion [55] and ProlificDreamer [78], we consider the following optimization problem to align the edited image, O, with the editing knowledge packed in a pretrained IP2P [6]:

$$\begin{aligned} O^* &= \arg\min_O \mathrm{D}_{\mathrm{KL}}\big(q_0^O(z_0|I, C_T) \| p_0(z_0|I, C_T)\big) \\ &= \arg\min_O \mathrm{D}_{\mathrm{KL}}\big(q_0^O(z_0|I) \| p_0(z_0|I, C_T)\big), \end{aligned} \tag{7}$$

where $\{q_t^O\}$ and $\{p_t\}$ are the forward (diffusion) and reverse (denoising) models in IP2P, respectively, $z_0 = \mathcal{E}(O)$ is the encoded output, and D_{KL} is the KL-divergence. As shown in ProlificDreamer [78], for each $t > 0$, we have

$$\begin{aligned} &\mathrm{D}_{\mathrm{KL}}\big(q_t^O(z_t|I) \;\|\; p_t(z_t|I, C_T)\big) = 0 \\ &\iff q_0^O(z_0|I) = p_0(z_0|I, C_T). \end{aligned} \tag{8}$$

By setting t to t_{rel}, we approximate the solution to the optimization problem in Eq. 7 with

$$\begin{aligned} O^* &= \arg\min_O \mathrm{D}_{\mathrm{KL}}\big(q_{t_{\mathrm{rel}}}^O(z_{t_{\mathrm{rel}}}|I) \| p_{t_{\mathrm{rel}}}(z_{t_{\mathrm{rel}}}|I, C_T)\big) \\ &= \arg\min_O \underbrace{\mathbb{E}_\epsilon\big[\log q_{t_{\mathrm{rel}}}^O(z_{t_{\mathrm{rel}}}|I) - \log p_{t_{\mathrm{rel}}}(z_{t_{\mathrm{rel}}}|I, C_T)\big]}_{\textbf{objective}(\mathfrak{L})}. \end{aligned} \tag{9}$$

By taking the gradient of the objective, $\mathfrak{L}$, with respect to O we get

$$\begin{aligned} \nabla_O \mathfrak{L} &= \nabla_O \mathbb{E}_\epsilon\big[\log q_{t_{\mathrm{rel}}}^O(z_{t_{\mathrm{rel}}}|I) - \log p_{t_{\mathrm{rel}}}(z_{t_{\mathrm{rel}}}|I, C_T)\big] \\ &= \mathbb{E}_\epsilon\big[\nabla_O \log q_{t_{\mathrm{rel}}}^O(z_{t_{\mathrm{rel}}}|I) - \nabla_O \log p_{t_{\mathrm{rel}}}(z_{t_{\mathrm{rel}}}|I, C_T)\big]. \end{aligned} \tag{10}$$

Now notice that since $s_\theta(z_t|I, C_T) \approx \nabla_{z_t} \log p_t(z_t|I, C_T)$, we have

$$\begin{aligned} &\nabla_O \log p_{t_{\mathrm{rel}}}(z_{t_{\mathrm{rel}}}|I, C_T) \approx s_\theta(z_{t_{\mathrm{rel}}}|I, C_T) \frac{\partial z_{t_{\mathrm{rel}}}}{\partial z_0} \frac{\partial z_0}{\partial O} \\ &= s_\theta(z_{t_{\mathrm{rel}}}|I, C_T) \alpha_{t_{\mathrm{rel}}} \frac{\partial \mathcal{E}(O)}{\partial O} \\ &= -\frac{\alpha_{t_{\mathrm{rel}}}}{\sigma_{t_{\mathrm{rel}}}} \epsilon_\theta(z_{t_{\mathrm{rel}}}, t_{\mathrm{rel}}, I, C_T) \frac{\partial \mathcal{E}(O)}{\partial O}. \end{aligned} \tag{11}$$

We rewrite $\nabla_O \log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I)$ as

$$\begin{aligned}
&\nabla_O \log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I) \\
&= \Big(\underbrace{\frac{\partial \log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I)}{\partial z_0}}_{\text{parameter score}} + \underbrace{\frac{\partial \log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I)}{\partial z_{t_{\text{rel}}}} \frac{\partial z_{t_{\text{rel}}}}{\partial z_0}}_{\text{path derivative}} \Big) \frac{\partial z_0}{\partial O} \\
&\approx \Big(\frac{\partial \log q^O_{t_{\text{rel}}}(z_{t_{\text{rel}}}|I)}{\partial z_{t_{\text{rel}}}} \alpha_{t_{\text{rel}}} \Big) \frac{\partial \mathcal{E}(O)}{\partial O} \\
&\approx -\frac{\alpha_{t_{\text{rel}}}}{\sigma_{t_{\text{rel}}}} \epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, \emptyset_T) \frac{\partial \mathcal{E}(O)}{\partial O}. \qquad (12)
\end{aligned}$$

The approximation in the third line of Eq. 12, where the parameter score is dropped, is motivated by *DreamFusion* [55], *Sticking-the-Landing* [59], and *Score Jacobian Chaining* [76]. Now, using Eq. 11 and Eq. 12, we can rewrite the gradient of the objective with respect to O in Eq. 10 as

$$\nabla_O \mathfrak{L} \approx \frac{\alpha_{t_{\text{rel}}}}{\sigma_{t_{\text{rel}}}} \mathbb{E}_\epsilon \Big[\underbrace{\epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, C_T) - \epsilon_\theta(z_{t_{\text{rel}}}, t_{\text{rel}}, I, \emptyset_T)}_{\text{signed relevance map}} \Big] \frac{\partial \mathcal{E}(O)}{\partial O}. \qquad (13)$$

Having the gradient of the objective of the edit with respect of the output, one can start with an initial output, $O_0 = I$, and iteratively update it to get $O_\infty = O^*$. According to Eq. 13, the magnitude of each update step is directly proportional to the absolute value of the relevance map; if the relevance map has a low value, the pixel would not be updated, and if the relevance map is higher, the corresponding pixel will change more significantly. In this paper, we explicitly limit the denoising process to only change the pixels with high relevance values, which is closely aligned with the theoretical procedure derived in this section.

We remark that our approach, which modifies the diffusion process rather than attempting an SDS-like optimization, is both more efficient and simpler (in that the optimization requires designing); further, the SDS-based approach entails certain approximations, such as the assumption on the parameter score, which may reduce performance. Empirically, we tested an implementation of the SDS-like optimization above, but found it produced images of worse quality. In theory, it may be possible to utilize this optimization-based approach instead; we leave additional investigations to future work.

F Relevance field ablation

We include here an ablation where 2D relevance maps are used *without* the 3D radiance field. By design, using the 3D radiance field has significantly better preservation of the original scene in edit-irrelevant areas, as measured by Image Sim. (↑), which increases from 0.83 to 0.89, and Edit PSNR (↑), which improves 1.6 dB (from 29.4 to 31). Importantly, however, the radiance field obtains this advantage with the semantic (e.g., Txt-Img Sim.) and image quality (NIQE)

metrics remaining unchanged. In Fig. 16 (rows 4-5; text *"give him sunglasses"*), the reason for these results can be seen in the noisier 2D maps, which highlight a greater proportion of irrelevant areas, compared to the highly localized and cleaner 3D radiance field renders. In other words, the radiance field improves localization without sacrificing performance along other dimensions, essentially by forcing the relevances to be consistent across views.

G Off-the-shelf segmentation ablation

Using a separate segmentation algorithm could be useful and/or perform similarly in certain cases. However, using the editing model itself to produce the mask has significant advantages. To this end, we also compare to a state-of-the-art segmentor, SAM-Track [10], capable of handling multiview inputs, with public models and code. The fundamental problem with this approach is that SAM struggles to segment content that *does not yet exist or is ambiguous*, i.e., SAM is not "edit-aware". For instance, in Fig. 16, we use SAM to extract masks for *"the area around the eyes that sunglasses would cover"*. However, this corresponds to an ill-defined image area, which covers the eyes and some surrounding areas, depending on the generated sunglasses. In Fig. 16, row two, we tried to isolate that region manually with points, but encounter difficulty since the sunglasses' boundaries do not exist in the unedited image. Similarly, text-based inputs (row three) oversegment. Our tests required non-trivial manual effort to fine-tune the points and text. In contrast, the 2D relevance maps are "edit-aware" and easily demarcate the target area; the 3D RF then combines these across views, to obtain cleaner localizations (see Fig. 16 and Fig. 10). Further, Fig. 17 shows SAM-Track outputs on two text prompts (upper row): since the hat does not yet exist in the image, SAM cannot localize it. We also show (bottom row) the results of NeRF editing with the SAM masks: we find it cannot add the hat, and instead only blurs his hair.

H Social impact

Since our "Watch Your Steps" is a generative model, it incurs the social issues of such models (e.g., data bias). Generative models, whether used for images, videos, 3D content, or other modalities, can perpetuate and even amplify existing biases present in the training data. This can lead to the reinforcement of harmful stereotypes and the exclusion of underrepresented groups. Additionally, the realistic outputs of these models can be misused to create deepfakes, which pose ethical concerns regarding privacy, consent, and misinformation. The potential for generative models to produce and disseminate misleading or harmful content necessitates careful consideration of their societal impact and the implementation of robust ethical guidelines and mitigation strategies.

Fig. 15: Sample rendered relevances from relevance fields trained on different scenes and different edit instructions. Each relevance field is visualized from multiple views, in addition to the corresponding views from the original NeRF model of the scene. Notice how each relevance field is mostly predominately around the region that is highly relevant to the edit. For example, in the face scene and with the instruction *"Give him blonde hair"*, only the hair is given high values in the field. This field allows localizing edits of the training views during each iterative update in a 3D consistent manner.

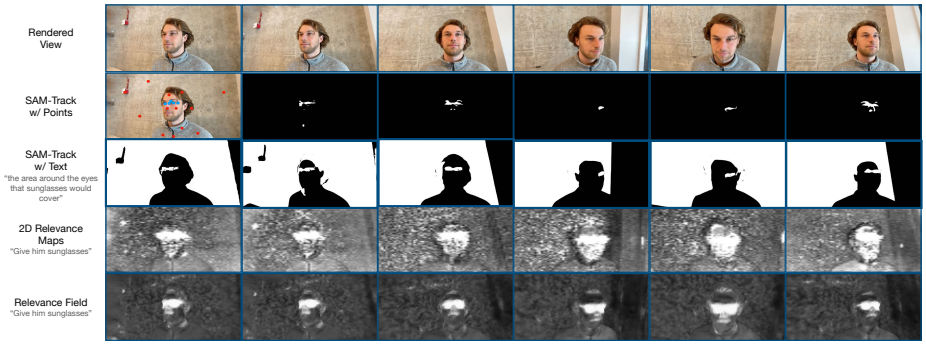

Fig. 16: Illustration of the rendered relevance maps from our relevance field, compared to independent 2D relevance maps and masks obtained using SAM-Track [10]. As visible in the results, the relevance field reduces the noise in the 2D relevance maps by enforcing a smoothness prior over the averaged-out values across different views. On the other hand, interactive/text-guided segmentation methods struggle to delineate regions that don't exist, such as the sunglasses in this scene.

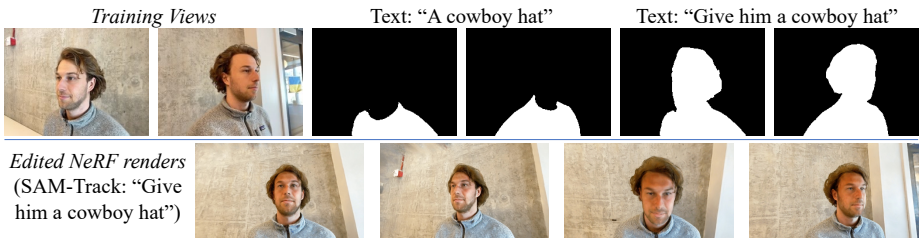

Fig. 17: SAM-Track [10] fails to produce masks for regions that do not exist (e.g., the "*cowboy hat*" in this case). This leads to the failure of the downstream editing task, where the hat was not successfully added to the scene.