Uniform Attention Maps: Boosting Image Fidelity in Reconstruction and Editing

Supplementary Material

Algorithm 1 Edit images with adaptive mask

Input: Given original image z_0, target prompt $\mathbf{c}_{tgt}$, source prompt $\mathbf{c}_{src}$, denoising model ϵ_θ, uniform cross-attention maps $\mathcal{C}$, null prompt $\mathbf{c}_\varnothing$, a dilation operation $dilate(\cdot)$.
$z_T^u \leftarrow \text{Invert}(z_0, \mathcal{C}, \mathbf{c}_\varnothing)$
$z_T^{src} \leftarrow \text{Invert}(z_0, \mathbf{c}_{src})$
$z_T^{tgt} \leftarrow z_T^{src}$
for $t = T$ to 1 **do**
  $\#\ Auxiliary\ Branch$
  $\epsilon_u \leftarrow \epsilon_\theta(z_t^u, \mathcal{C}, \mathbf{c}_\varnothing)$
  $\hat{z}_{0,t}^u \leftarrow \frac{1}{\sqrt{\alpha_t}} z_t^u - \frac{1-\alpha_t}{\sqrt{\alpha_t}} \epsilon_u$
  $\#\ Source\ Branch$
  $\epsilon_{src} \leftarrow \epsilon_\theta(z_t^{src}, \mathbf{c}_{src})$
  $\hat{z}_{0,t}^{src} \leftarrow \frac{1}{\sqrt{\alpha_t}} z_t^{src} - \frac{1-\alpha_t}{\sqrt{\alpha_t}} \epsilon_{src}$
  $\#\ Target\ Branch$
  $\epsilon_{tgt} \leftarrow \epsilon_\theta(z_t^{tgt}, \mathbf{c}_{tgt})$
  $\hat{z}_{0,t}^{tgt} \leftarrow \frac{1}{\sqrt{\alpha_t}} z_t^{tgt} - \frac{1-\alpha_t}{\sqrt{\alpha_t}} \epsilon_{tgt}$
  $M \leftarrow dilate(|\hat{z}_{0,t}^{tgt} - \hat{z}_{0,t}^{src}| \leq \lambda)$
  if $t < T_{mask}$ **then**
    $\hat{z}_{0,t}^{tgt} \leftarrow M \odot \hat{z}_{0,t}^u + (1 - M) \odot \hat{z}_{0,t}^{tgt}$
  end if
  $z_{t-1}^{tgt} \leftarrow \sqrt{\alpha_{t-1}} \hat{z}_{0,t}^{tgt} + \sqrt{1-\alpha_{t-1}} \epsilon_{tgt}$
  $z_{t-1}^{src} \leftarrow \sqrt{\alpha_{t-1}} \hat{z}_{0,t}^{src} + \sqrt{1-\alpha_{t-1}} \epsilon_{src}$
  $z_{t-1}^u \leftarrow \sqrt{\alpha_{t-1}} \hat{z}_{0,t}^u + \sqrt{1-\alpha_{t-1}} \epsilon_u$
end for
return z_0^{tgt}

A. Adaptive Mask-Guided Image Editing: Algorithm Overview

The pseudocode for our adaptive mask method is shown in Algorithm 1. The algorithm takes an input image z_0, a target prompt $\mathbf{c}_{tgt}$, and a source prompt $\mathbf{c}_{src}$. The method starts by inverting the image through auxiliary and source branches and then initializes the target branch from the source branch.

At each timestep t, we compute noise predictions and update the latent variables in the auxiliary, source, and target branches. It generates an adaptive mask M by comparing the clean images $\hat{z}_0$ from the target and source branches and applies a dilation operation to ensure robustness. The mask M is then used to blend the predictions from the auxiliary and target branches, preserving key details of the original image while applying the edits.

The process repeats until the final image z_0^{tgt} is returned, incorporating the original information and the desired modifications.

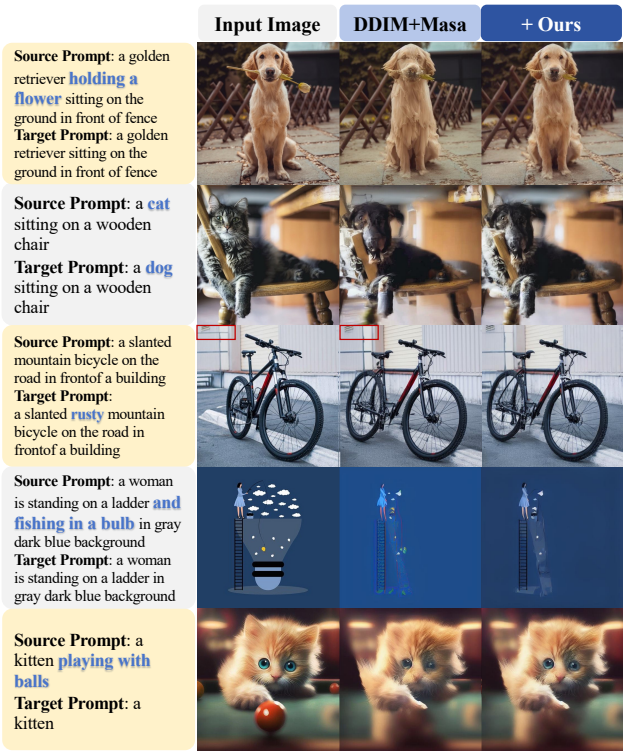

Figure 10. More examples of image editing on the PIE benchmark. Examples of image editing on the PIE benchmark, comparing the DDIM+Masa method with our image editing method.

B. More Examples of Image Reconstruction

Figs. 11 to 14, provide additional examples of image reconstruction using DDIM inversion with 20 timesteps on the PIE benchmark, showcasing the performance of our method in comparison to null prompts and source prompts. In Figs. 11 to 14, we observe the reconstruction of various images. The results using the null prompt often produce blurred or incorrect outputs, while the source prompt reconstructions are better but still show visible artifacts. By leveraging uniform attention maps, our method demonstrates significant improvements, yielding clearer and more accurate reconstructions that align closely with the original input images, preserving important details such as texture and shape. These examples confirm the robustness of our approach across different image types, showing that our method consistently outperforms the baseline approaches in generating high-quality reconstructions that faithfully resemble the input images.

C. More Examples of Image Editing

Fig. 10 showcases the effectiveness of our image editing method compared to the DDIM+Masa baseline. Our method consistently produces more accurate, detailed, and visually coherent edits across various scenarios, such as transforming animals, modifying complex objects, and retaining structural fidelity in abstract compositions, outperforming the baseline in terms of both precision and consistency.

D. More Experimental Details

Visualize Experiment Details. We conduct experiments in Fig. 3 and Fig. 2 using Stable Diffusion v1.4 with DDIM inversion and reconstruction under 20 inference steps. At each timestep, the cross-attention term $A^{(l)}$ is extracted from U-Net layers with an output dimension of 64×64. The clean predicted image $\hat{z}_{0,t}$ is also generated at each timestep t to evaluate the reconstruction fidelity.

In Fig. 3, the Mean Squared Error of the cross-attention term is computed at the pixel level as the discrepancy between $A^{(l)}_{\text{inv}}$ and $A^{(l)}_{\text{rec}}$, with the results averaged across all pixels. Similarly, the reconstruction error is calculated as the pixel-level MSE between the predicted clean images $\hat{z}_{0,\text{inv}}$ and $\hat{z}_{0,\text{rec}}$. These two MSE metrics are aggregated across all timesteps for each image. The scatter plot in Fig. 3 illustrates a strong positive correlation between the cross-attention discrepancies and the reconstruction errors, demonstrating that misalignment in the cross-attention mechanism is a significant contributor to the errors in the final reconstructed images.

In Fig. 2, the extracted cross-attention terms $A^{(l)}$ are visualized as heatmaps to show their temporal evolution across the inversion and reconstruction processes. Fig. 2 (a) highlights the discrepancies in the cross-attention maps under source prompts, null prompts, and our proposed method. The heatmaps for the source and null conditions reveal significant misalignments between the inversion and reconstruction phases, emphasized by the black-boxed regions. In contrast, our method ensures consistent cross-attention alignment throughout the process. Furthermore, Fig. 2 (b) presents the corresponding clean predicted images $\hat{z}_{0,t}$ at various timesteps, showing that the proposed method maintains high-quality reconstructions, while the source and null prompts result in noticeable distortions.

Experimental Metrics. The primary goal of semantic image editing is to accurately modify specific objects or scenes in an image as described in the target text. This process ensures that only the intended part of the image is altered while retaining unmodified parts as much as possible. To assess the effectiveness of our methods, we utilize metrics from prior work [14]. We report the following metrics: (1) Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE): These metrics evaluate the faithfulness of the generated images by comparing them to the input images. (2) LPIPS [37]: LPIPS is a deep learning-based metric that assesses perceptual similarity between images, aligning more closely with human perception than traditional metrics. (3) SSIM [34]: SSIM measures the similarity between the two images, focusing on changes in structural information, luminance, and contrast. (4) CLIP Score [26]: We employ a combination of CLIP image and text models to calculate the similarity between generated images and corresponding texts, measuring the alignment between the generated image and the target text. We report CLIP Score for both the entire image (Whole) and within the editing mask (Edited), where regions outside the mask are blacked out. (5) Structural Distance [31]: This metric assesses structural changes in images.

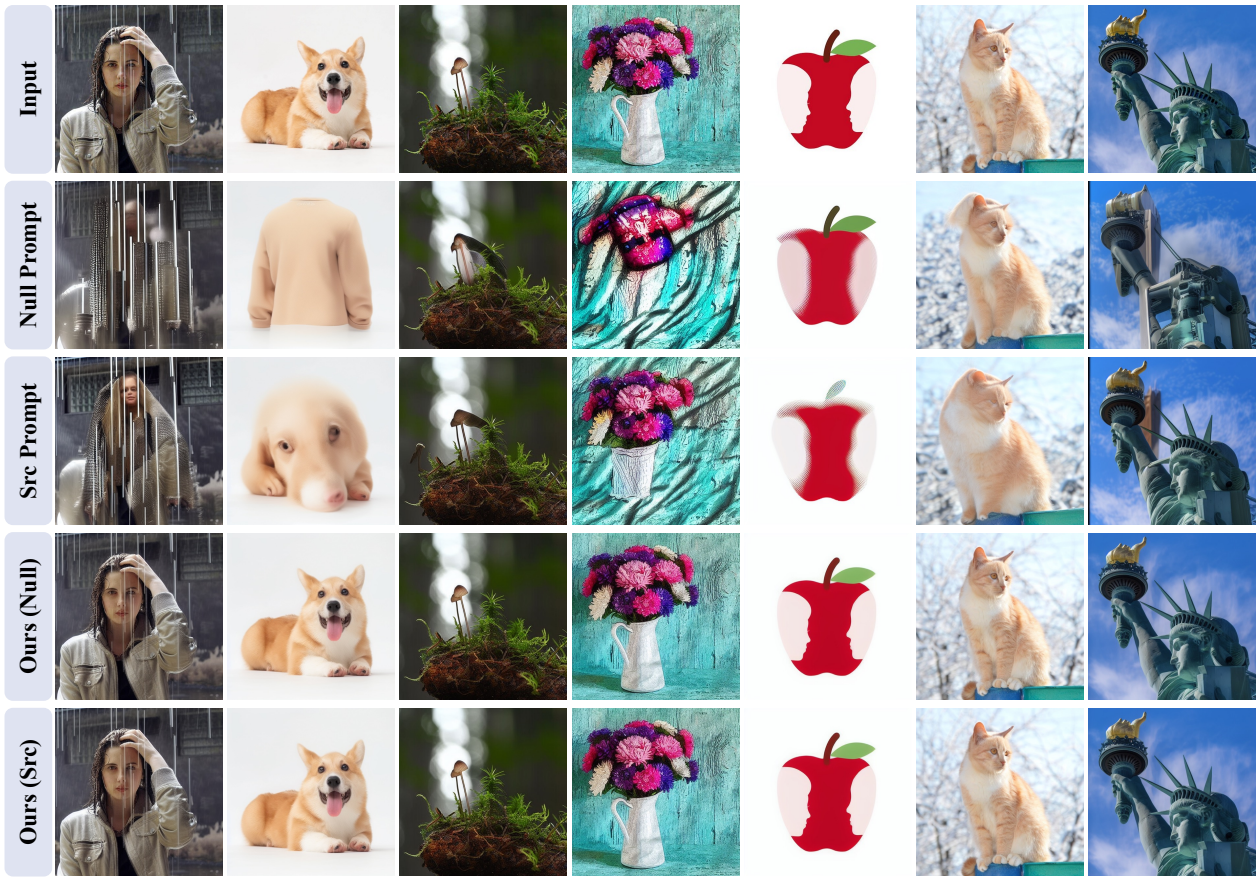

Figure 11. Examples of image reconstruction on the PIE benchmark. The first row shows the input images. The second and third rows display the results using a null prompt (an empty string) and a source prompt from the benchmark, respectively. The fourth and fifth rows show the results from our method with different value tokens, demonstrating superior reconstruction quality and better alignment with the original input images.

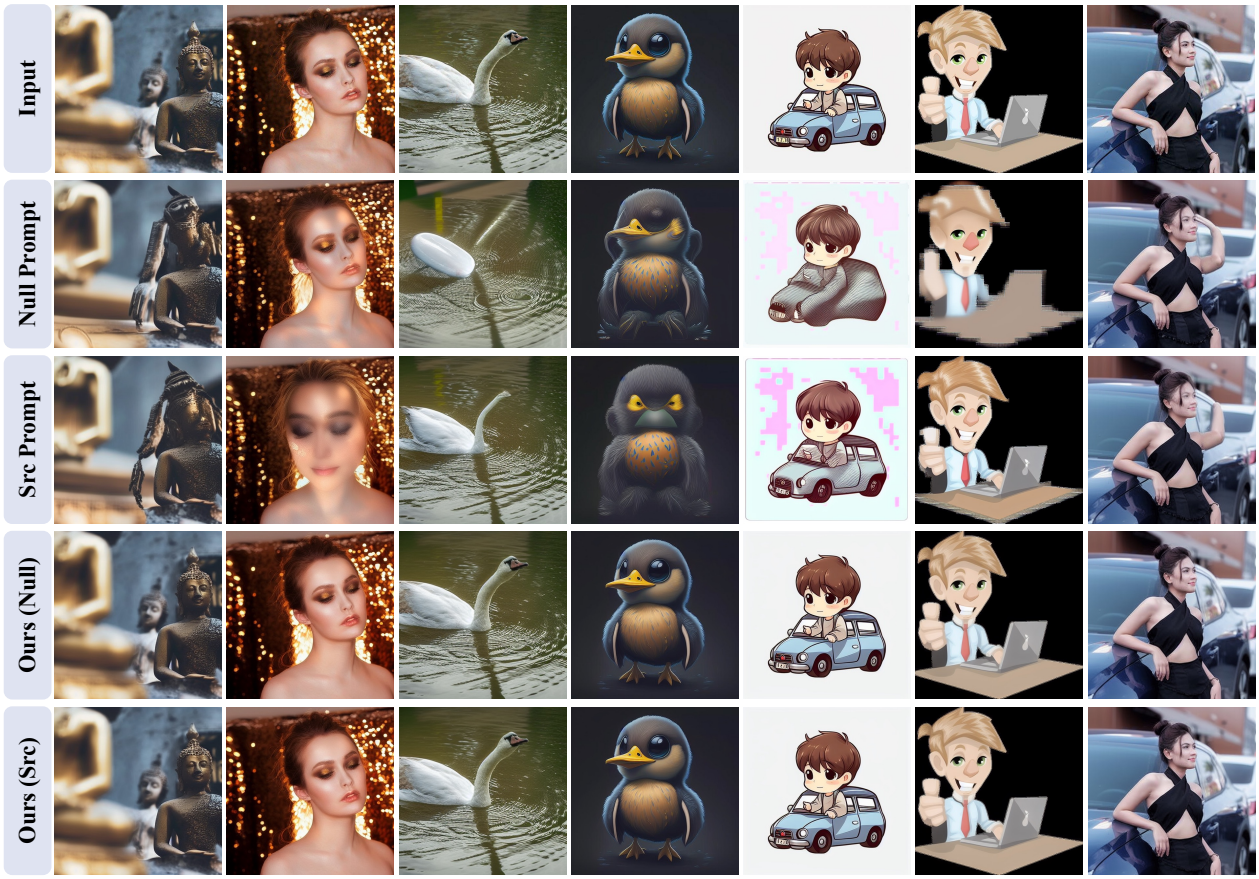

Figure 12. More examples of image reconstruction on the PIE benchmark.

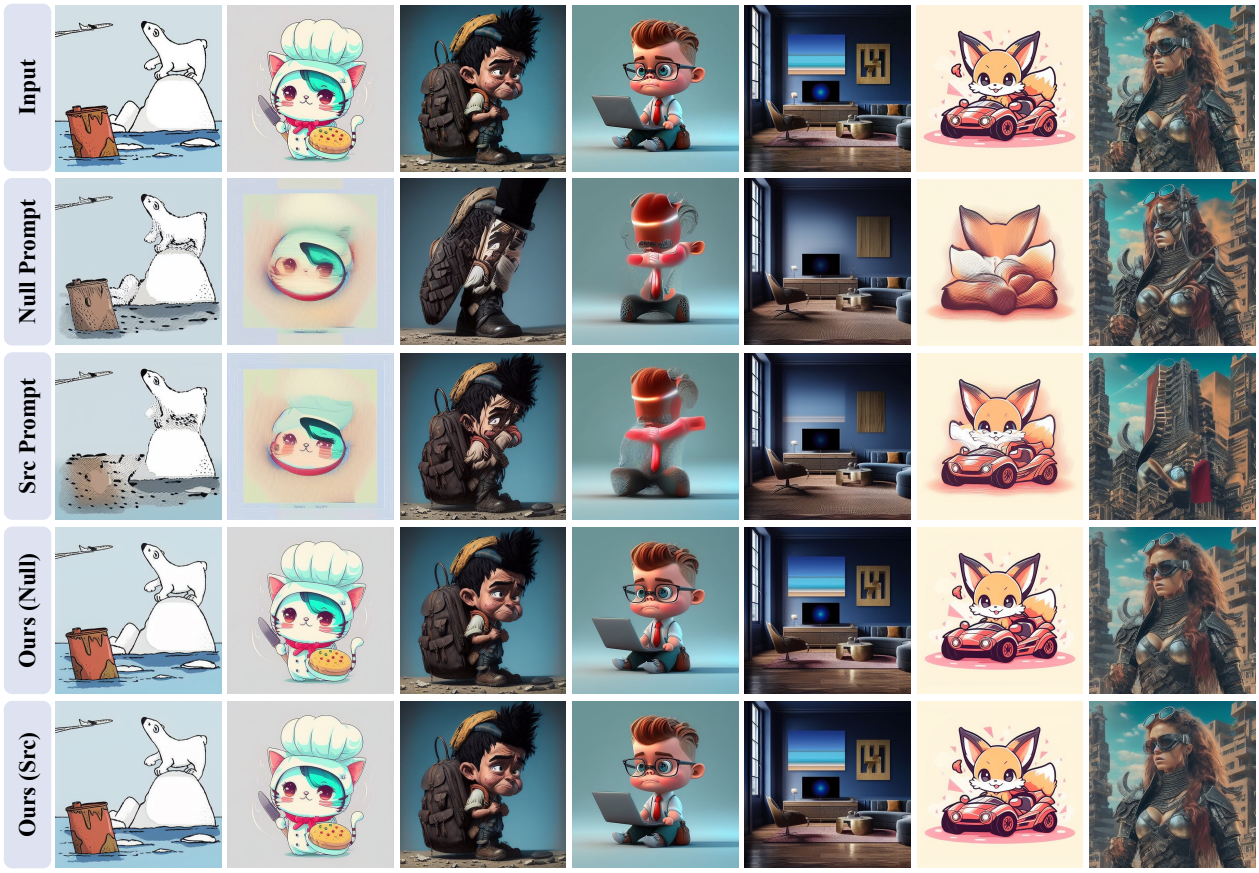

Figure 13. More examples of image reconstruction on the PIE benchmark.

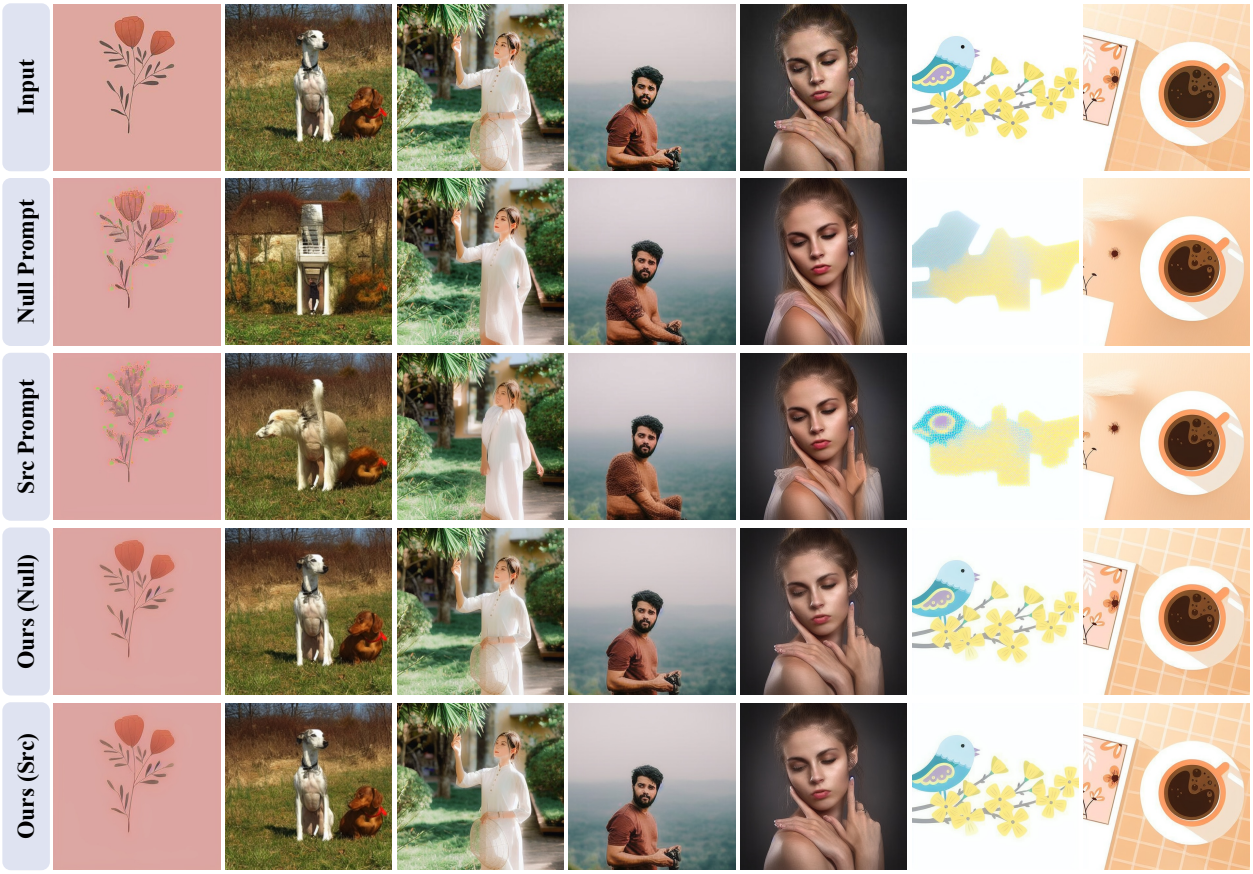

Figure 14. More examples of image reconstruction on the PIE benchmark.