Edge-Aware Image Manipulation via Diffusion Models with a Novel Structure-Preservation Loss

Supplementary Material

Contents

A. Additional Experiments and Analyses

A.1. Comparison with Task-Specific Structure-Preserving Methods

In addition to LDM-based editing models, we compare our method against task-specific methods across several key structure-preserving tasks.

Photo-realistic Style Transfer synthesizes images by merging content and style from separate images. We compare our method with PCAKD [9], utilizing 60 content-style image pairs from the DPST dataset [39]. The evaluation is performed using prompts generated via the method described in Appendix E. In Fig. 1, our method better captures subtle stylistic features compared to PCAKD [9], producing higher-quality results.

Image Harmonization adjusts a foreground object's color and brightness to match a composite image's background. Similar to style transfer, we derive prompts using GPT-4o [43]. In Fig. 2, our method yields results that blend more consistently with the background than PCTNet [16].

Image Tone Adjustment modifies the brightness, contrast, and color balance of the input image. We compare our method with CLIPtone [31] using a subset of [5], which is a test set consisting of approximately 500 images and around 50 different tone descriptions. The source and edit text prompt pairs are constructed in the format "a normal photo of..." → "a [tone] photo of..." by combining image captions generated by BLIP [35] and tone descriptors. Compared to CLIPtone [31], our method more accurately and naturally achieves the intended tone.

Seasonal change alter environmental contexts. We compare with CycleGAN's [78] pre-trained summer-to-winter model, using approximately 550 provided test set images. The source and edit text prompt pairs follow the format "a photo of summer ..." ↔ "a photo of winter ...". Fig. 4 demonstrate our method maintains pixel-level structures and achieves better edit quality compared to CycleGAN, thanks to our diffusion-based prior.

Time-lapse Editing alter temporal contexts. We evaluate against Pix2pix [22], using its pre-trained day-to-night model. We use the night-to-day dataset of [29], and use 350 daytime images. The source and edit text prompt pairs are manually created in the form "a photo of ... at day" → "a photo of ... at night". Fig. 5 demonstrate our method maintains pixel-level structures and achieves better edit quality compared to Pix2pix thanks to our diffusion-based prior.

Quantitative Comparison. For the evaluation, we use the same metrics used in Sec. 4.1. As shown in Tab. 1, our model achieves superior prompt fidelity compared to methods specifically designed for each structure-preserving image editing task, with notable advantages in structural preservation through our SPL. LPIPS, being a perceptual metric, may be lower for our model compared to CLIPtone due to our stronger emphasis on structure preservation rather than perceptual similarity alone. As discussed in Sec. 4.3, we note that SSIM scores can be misleading in tasks involving significant brightness changes (e.g., image tone adjustment or time-lapse), since SSIM strongly penalizes brightness variations.

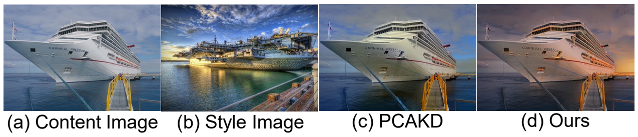

(a) Content Image (b) Style Image (c) PCAKD (d) Ours

Figure 1. **Qualitative comparison on photorealistic style transfer**. Given a content image (a) and a style image (b), our method effectively transfers the sunrise style, producing a naturally stylized result (d). In contrast, PCAKD (c) fails to transfer this style effectively.

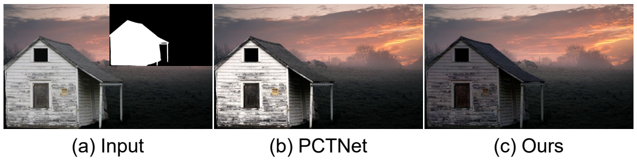
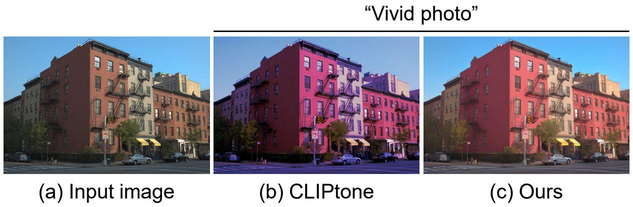

(a) Input (b) PCTNet (c) Ours

Figure 2. **Qualitative comparison on image harmonization**. (a) Composed image with foreground mask (top-right). The Result from PCTNet (b) exhibits clear lighting inconsistencies. Our method (c) seamlessly blends the foreground and background.



(a) Input image (b) CLIPtone (c) Ours

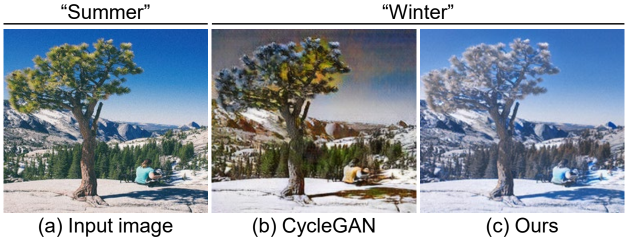
Figure 3. **Qualitative comparison on tone adjustment**. While the result of CLIPtone (b) is overly saturated, our method (c) produces a naturally-toned result well-aligned with the text description.

A.2. Qualitative Comparison on the AnyEdit Benchmark

We provide qualitative comparisons on the AnyEdit benchmark [72] in Fig. 6 and Fig. 7 further support these findings. The visualizations highlight our method's ability to preserve fine-grained, pixel-level edge structures, whereas competing methods often introduce noticeable structural artifacts or fail to fully respect the source image's pixel-level edge structures. Overall, these results strongly corroborate the conclusions drawn from our main experiments on the PIE-Bench subset.

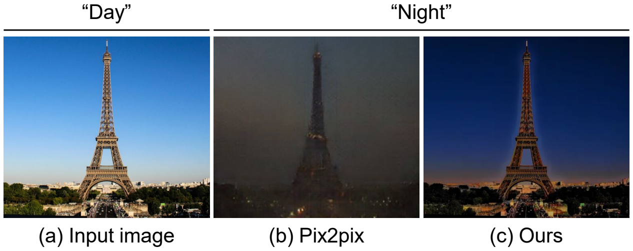
A.3. Cross Attention Mask Upsampling with Different Backbones

Our cross-attention mask upsampling method extends naturally to various diffusion model architectures beyond the original LDM [53], such as SDXL [49] and FLUX [1]. For



(a) Input image (b) CycleGAN (c) Ours

Figure 4. **Qualitative comparison on season change**. While CycleGAN (b) produces an incoherent result, our method (c) successfully generates a natural seasonal transformation, realistically integrating effects like snow.



(a) Input image (b) Pix2pix (c) Ours

Figure 5. **Qualitative comparison on time-lapse**. Pix2Pix (b) introduces significant structural distortions. Our method (c) achieves a realistic time-of-day change while maintaining the structure.

Method	Preservation			Prompt fidelity	
	SPL($\times 10^2$) ↓	SSIM↑	LPIPS↓	CLIP S.↑	CLIP D.↑
Photorealistic Style Transfer					
Ours	**0.006**	**0.879**	**0.182**	**0.254**	**0.147**
PCAKD	0.752	0.478	0.346	0.248	0.130
Image Tone Adjustment					
Ours	**0.010**	0.680	0.250	**0.237**	**0.078**
CLIPTone	0.016	**0.862**	**0.154**	0.226	0.075
Time-lapse					
Ours	**0.070**	0.240	**0.471**	0.234	**0.192**
Pix2pix	0.534	**0.396**	0.623	**0.235**	0.136
Season/Weather Change					
Ours	**0.061**	**0.856**	**0.266**	0.197	**0.170**
CycleGAN	1.358	0.463	0.350	**0.198**	0.097

Table 1. **Quantitative comparison with task-specific editing methods.** Our method consistently achieves the best structure preservation loss across all tasks while maintaining high prompt fidelity. Note that SSIM is sensitive to luminance and contrast variations; thus, for tasks requiring brightness or contrast adjustments (e.g., tone adjustment, time-lapse), SSIM scores may not accurately reflect structural preservation. Additionally, LPIPS primarily captures perceptual similarity rather than structural fidelity.

[1] https://huggingface.co/black-forest-labs/FLUX.1-dev

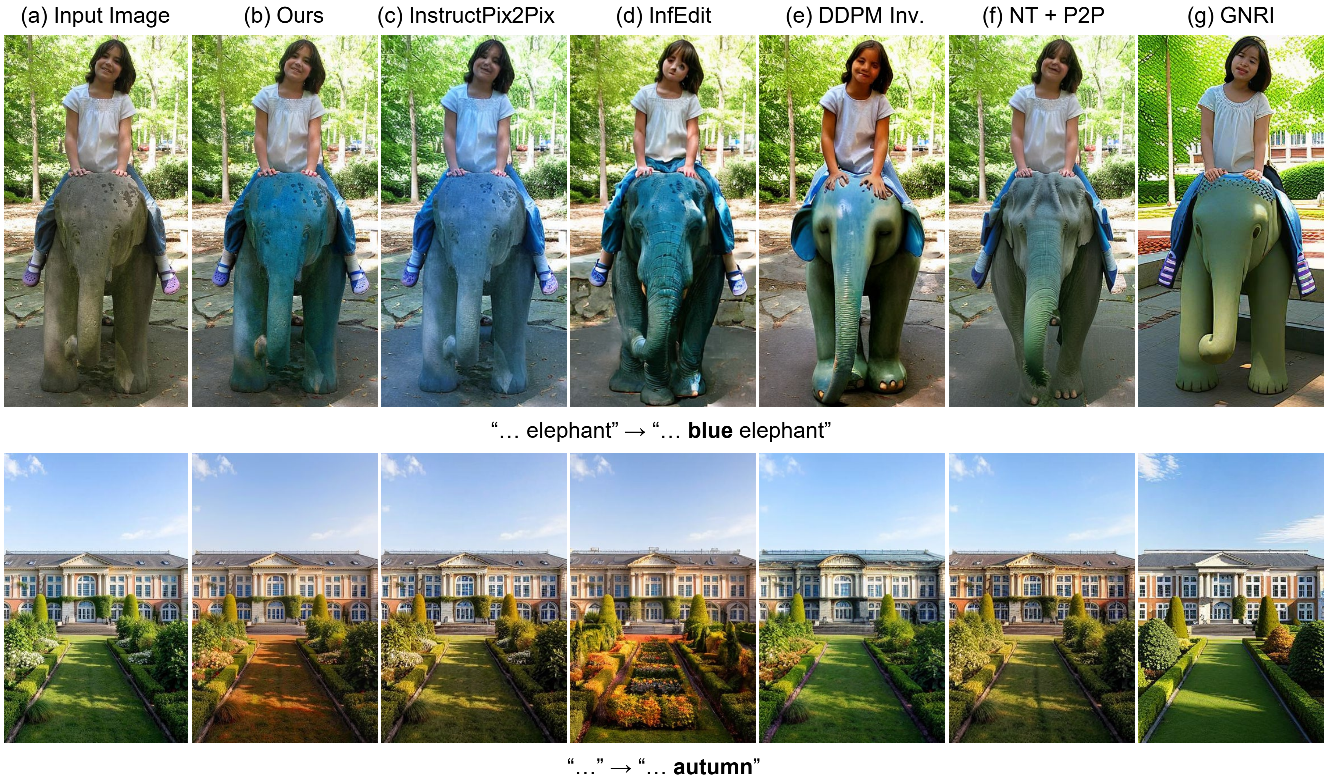


Figure 6. **Qualitative comparison on global editing tasks.** Our method (b) successfully applies the edit while preserving fine-grained structural details. Other methods (c-g) exhibit either low prompt fidelity or significant structural distortions

U-Net based backbones like SDXL, we follow Prompt-to-Prompt [20] and extract the cross-attention maps of resolution 32×32 from the bottleneck layer. For FLUX, which has a DiT [46] structure, we extract the average attention map of resolution 64×64 from intermediate blocks 12 through 18. Our guided upsampling algorithm then refines these coarse initial maps to the model's native output resolution—from 16×16 to 512×512 for the original LDM, and 32×32 to 1024×1024 for SDXL and 64×64 to 1024×1024 FLUX. This demonstrates the flexibility of our approach in adapting to different architectures. We show qualitative results for SDXL in Fig. 8 and for FLUX in Fig. 9, respectively.

A.4. Failure Case Analysis

Cross-Attention Mask Upsampling As demonstrated in Fig. 10, our guided upsampling technique is generally effective at capturing object silhouette with high fidelity. However, the accuracy of the final mask is fundamentally dependent on the quality of the initial coarse cross-attention map. In some cases, if the initial map is not well-localized, the refined mask may cover a slightly larger region than the target object, as shown in the red box of Fig. 10. When this occurs, SPL is inadvertently applied to this slightly oversized region. While the resulting edit still adheres to the prompt, this can cause subtle edge structures from the source image to be preserved.

Structure Preservation Loss The limitation of SPL becomes apparent in tasks that are inherently ambiguous for a structure-preserving method, where an edit may not be entirely aligned with the user's intention. We demonstrate this with a challenging material editing task: transforming a leather bag into a denim one, shown in Fig. 11. As intended, SPL successfully preserves the bag's overall macro-structure, such as the arm strap and its buckles, and also the fine-grained edge details of the original material's texture. The final edited result is coherent and appears natural with appropriate shading. However, if a user intends a structure-breaking material change that completely replaces the micro-texture, such an edit falls outside the intended scope of SPL.

A.5. Scheduling of Attention Conditioning and Structure Preservation Loss.

We explore how the scheduling of attention conditioning and structure preservation loss affects edits. As we can see in Fig. 13, keeping the attention conditioning strengthens coarse structure preservation, but even with full attention conditioning, the fine structural details of the input image are distorted. However, when combined with structure

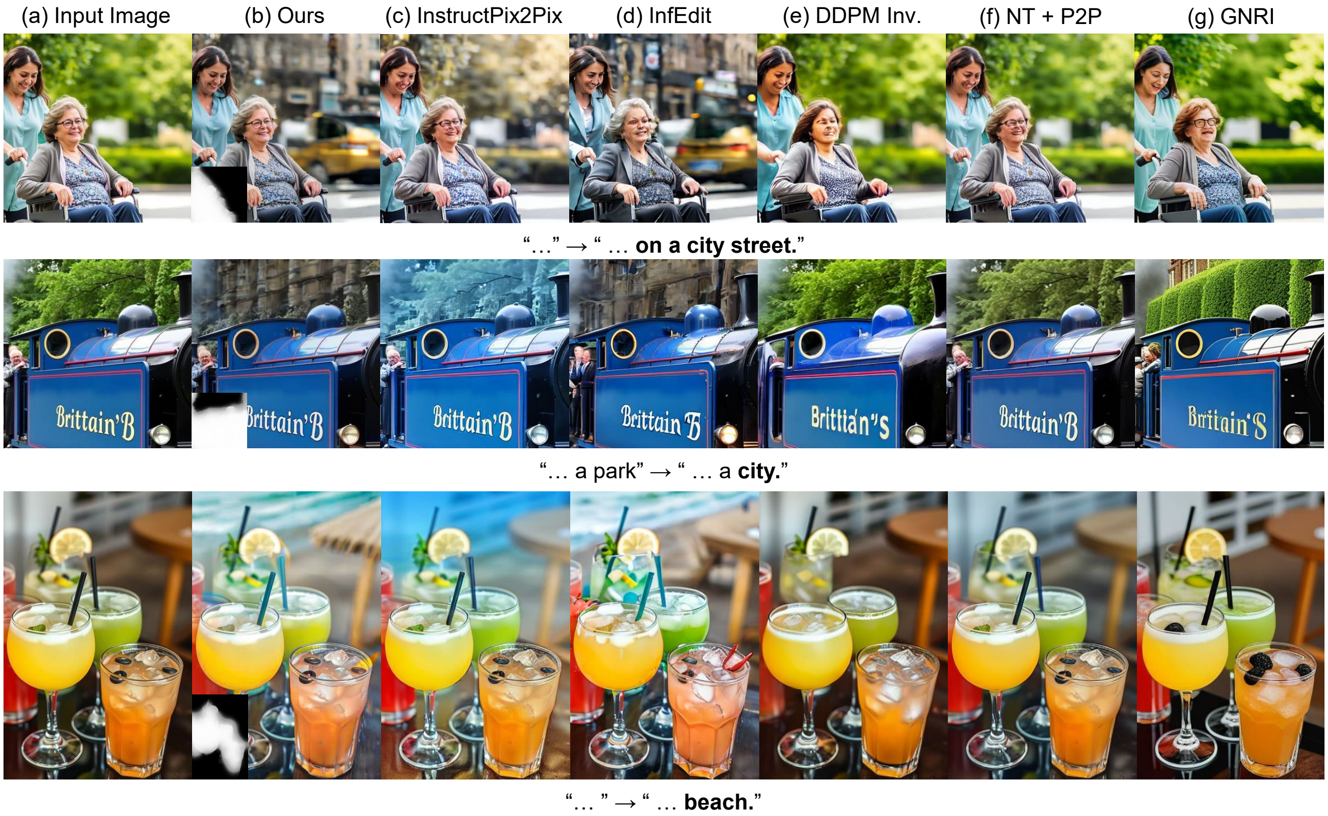


Figure 7. **Qualitative comparison of local editing tasks.** Our method can generate an edit mask from the text prompt (b, bottom-left) to enable precise local editing. Other methods (c-g) fail to preserve the structure of the content shared between the source and target prompts.

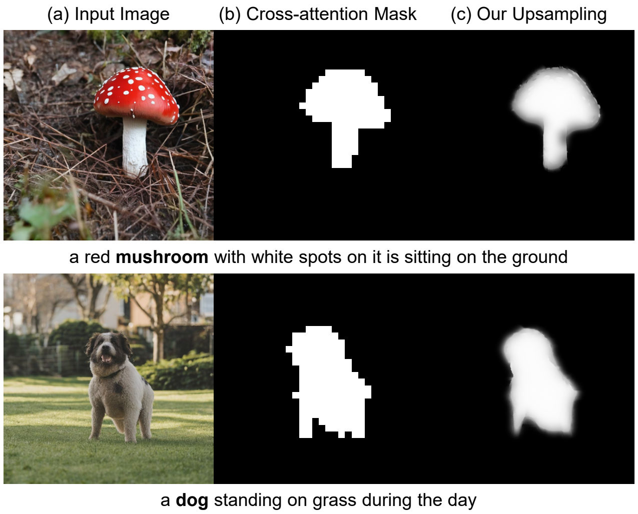

(a) Input Image
(b) Cross-attention Mask
(c) Our Upsampling
a red **mushroom** with white spots on it is sitting on the ground
a **dog** standing on grass during the day

Figure 8. **Cross-attention mask upsampling for SDXL [49].** By upsampling the coarse attention map (b), our method generates a sharp, high-resolution mask (c).

preservation loss, we can see that the pixel-level edge structural fidelity is kept.

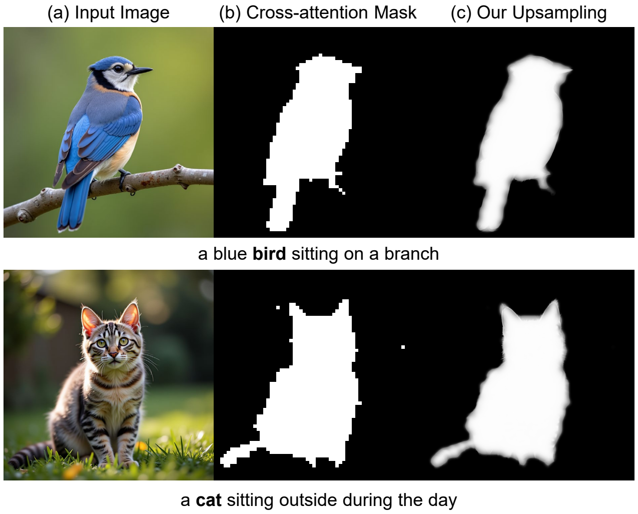
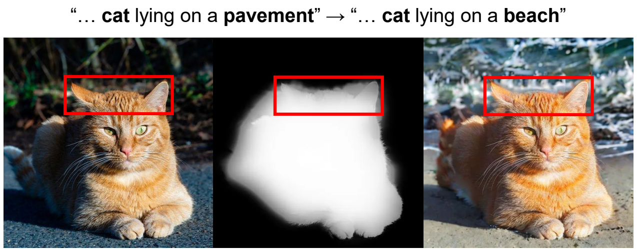

(a) Input Image
(b) Cross-attention Mask
(c) Our Upsampling
a blue **bird** sitting on a branch
a **cat** sitting outside during the day

Figure 9. **Cross-attention mask upsampling for FLUX.** By upsampling the coarse attention map (b), our method generates a sharp, high-resolution mask (c).

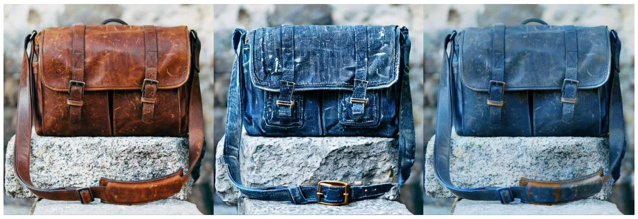
A.6. Validation as a Structural Difference Metric

To validate our proposed loss as a robust metric for structural similarity, we evaluate its response to a range of im-


“… **cat** lying on a **pavement**” → “… **cat** lying on a **beach**”

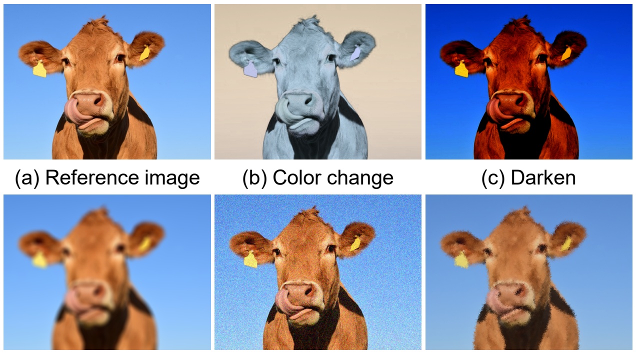
(a) Input image (b) Mask (c) Ours

Figure 10. **Imprecise Edit Mask Example.** The soft boundaries of the upsampled mask (b) can sometimes extend slightly beyond the foreground object. As a result, subtle structural details from the source image (a) are unintentionally preserved near the cat’s silhouette (c).



(a) Input image (b) Baseline (c) Ours

Figure 11. **Material editing example from leather to denim**. Our method (c) preserves the fine-grained texture and wear patterns of the original leather, while the baseline (b) breaks the structure and replaces the material entirely.



(a) Reference image (b) Color change (c) Darken

(d) Lens blur (e) Gaussian noise (f) Jitter

Figure 12. **Examples of image distortions.** (b-c) Non-structural distortions, (d-f) Structural distortions.

age distortions as seen in Fig. 12. The results in Tab. 2 demonstrate that our metric successfully disentangles structure from appearance. It registers a low penalty for non-structural distortions (e.g., color and brightness shifts) while correctly identifying structural distortions. In contrast, common metrics like SSIM [65] and LPIPS [75] often conflate these two, assigning high penalties to non-structural changes.

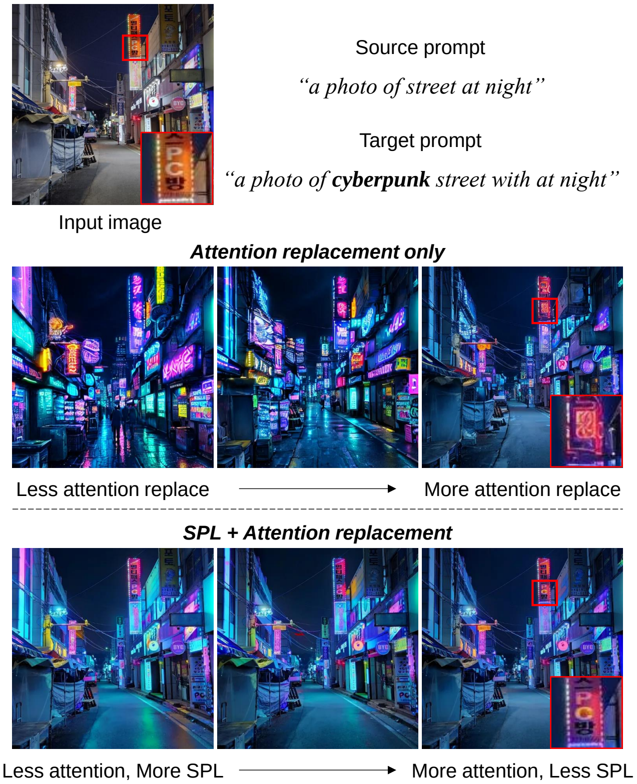

Source prompt
“a photo of street at night”
Target prompt
*“a photo of **cyberpunk** street with at night”*
Input image
Attention replacement only
Less attention replace
More attention replace
SPL + Attention replacement
Less attention, More SPL
More attention, Less SPL

Figure 13. **Scheduling of attention conditioning and structure preservation loss.** We analyze the impact of attention conditioning and the structure preservation loss on structural fidelity. As shown in Fig. 13, applying attention conditioning throughout the generative process helps retain coarse structures but fails to preserve fine details, even at full conditioning. In contrast, incorporating our structure preservation loss effectively maintains pixel-level edge fidelity, even with reduced attention conditioning.

Distortion	**SPL** ($\times 10^2$)↓	**PSNR**↑	**SSIM**↑	**LPIPS**↓
Color change	0.080	11.06	0.927	0.504
Darken	0.063	10.60	0.664	0.195
Lens blur	0.241	24.77	0.776	0.301
White noise	0.356	20.56	0.338	0.551
Jitter	0.325	22.20	0.783	0.196

Table 2. **Comparison of image similarity metrics across different types of distortions.**

A.7. Generalization to Different Backbones

To demonstrate the generality and modularity of our approach, we apply our structure-preserving editing method to a different baseline: the combination of Null-text inversion [41] and Prompt-to-Prompt [20]. As shown in Fig. 14, our method enhances the structural fidelity of the baseline’s output, confirming its effectiveness across different editing frameworks.

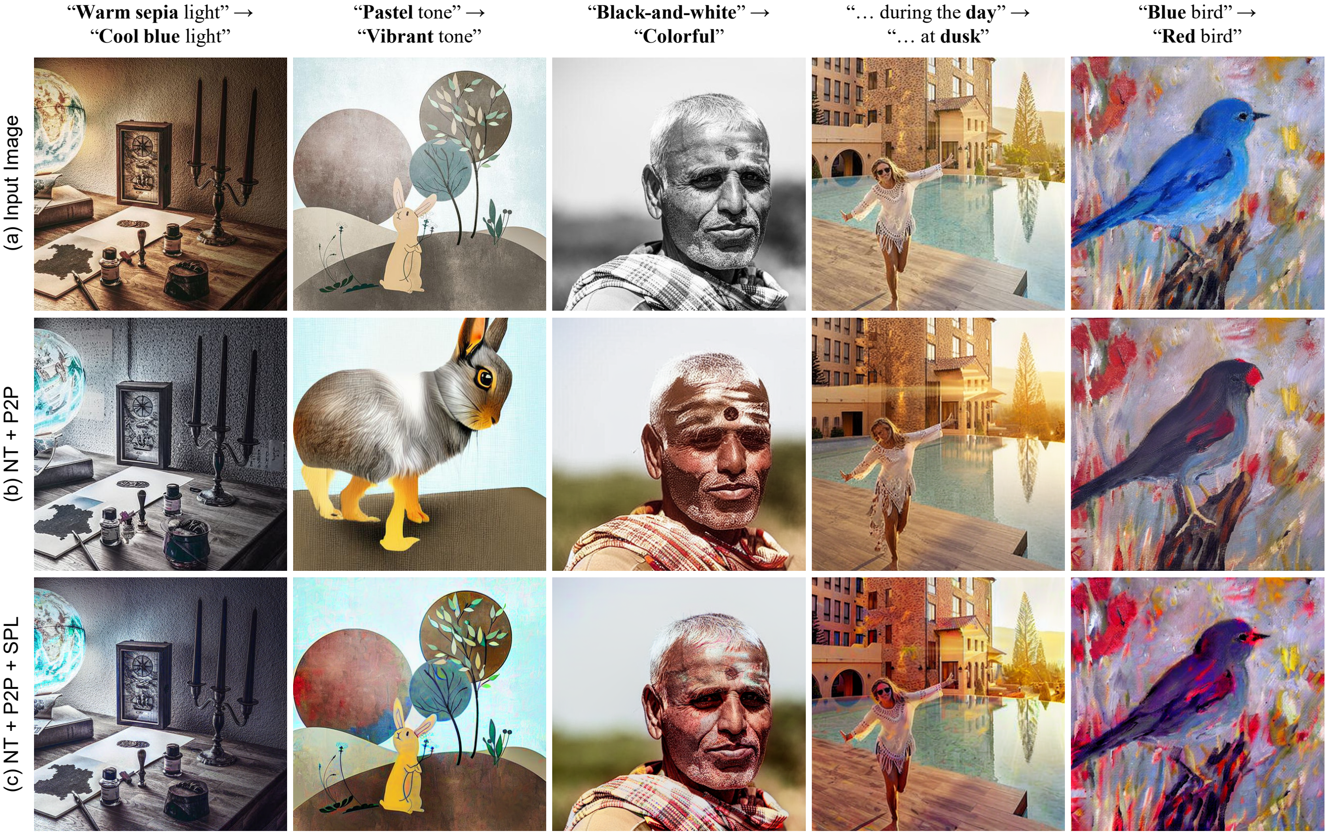

"**Warm sepia** light" → "**Cool blue** light"
"**Pastel** tone" → "**Vibrant** tone"
"**Black-and-white**" → "**Colorful**"
"… during the **day**" → "… at **dusk**"
"**Blue** bird" → "**Red** bird"
(a) Input Image
(b) NT + P2P
(c) NT + P2P + SPL

Figure 14. **Generalizability of our method across different baseline model.** Our method can be integrated into diverse LDM-based image editing pipelines (e.g., Null-text inversion + Prompt-to-Prompt), enhancing their ability to preserve the structural details of the input image during editing.

A.8. Additional Qualitative Comparison

We provide additional qualitative comparison results with LDM-based editing methods in Fig. 15. We can see that our method persistently achieves the best pixel-level edge structure preservation without the loss of prompt fidelity.

B. Additional Implementation Details

For all experiments, we use total inference steps of $T = 15$. For the structure preservation loss, defined via a local linear model in Equation Eq. (1), we configure a window size of $\omega_k = 11$ and a regularizer $\epsilon = 10^{-4}$. In the optimization-driven denoising process, we apply stochastic gradient descent with a learning rate $\eta = 1$ and momentum 0.9. During optimization we use $\lambda = 10^{-4}$ for to emphasize structural fidelity (Eq. (10)). We fix the number of optimization iterations at $k = s = 100$ and structure preservation loss threshold timestep $t_{\text{SPL}} = 12$. For tasks requiring localized edits, we employ the upscaled masks from Sec. 3.3. All experiments were conducted on an NVIDIA A6000 GPU. The models used in our experiments are as follows: for InfEdit [67], we used LCM Dreamshaper v7; for Instruct-Pix2Pix [4], the official model provided by the authors was employed. Both DDPMInv [21] and NT+P2P [20, 41] were implemented using Stable Diffusion v1.4.

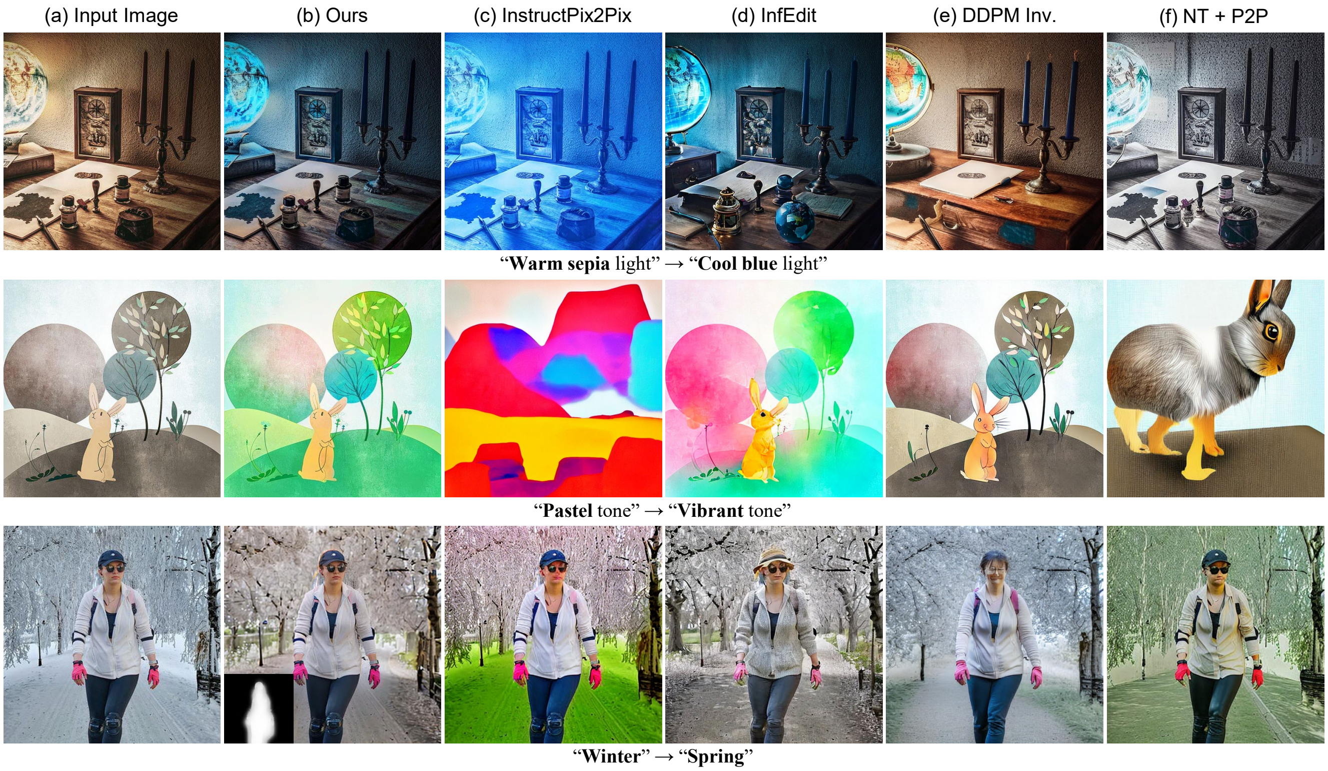
When applying the coarse structure preservation through attention conditioning as detailed in Sec. 3.2, we observe that applying this attention conditioning across all denoising timesteps can overly constrain the latent, reducing fidelity to the edit prompt p_{edit}. To balance coarse structure preservation with edit flexibility, we schedule the attention conditioning, applying f_t^{src} only for timesteps $t \geq t_{\text{attn}}$. We set this attention conditioning scheduling timestep as $t_{\text{attn}} = 12$.

C. Derivation of the LLM Coefficients

Given images I^E and I^S, the local linear model defines the relationship between these two images within a local window ω_k as:

$$I_i^S = a_k\, I_i^E + b_k, \quad \forall i \in \omega_k, \tag{12}$$

We derive the coefficients a_k and b_k by minimizing the



"**Warm sepia** light" → "**Cool blue** light"

"**Pastel** tone" → "**Vibrant** tone"

"**Winter**" → "**Spring**"

Figure 15. **Additional qualitative comparison with LDM-based image editing methods**. The first and second rows demonstrate global editing results. The Last row shows additional local editing results.

cost function $E(a_k, b_k)$ within a local window ω_k:

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left((a_k I_i^E + b_k) - I_i^S\right)^2. \tag{13}$$

The minimum is found by setting the partial derivatives with respect to a_k and b_k to zero.

Derivation of the Offset Coefficient b_k Differentiating E with respect to b_k and setting the result to zero yields:

$$\frac{\partial E}{\partial b_k} = 2 \sum_{i \in \omega_k} (a_k I_i^E + b_k - I_i^S) = 0$$
$$\implies a_k \sum_{i \in \omega_k} I_i^E + \sum_{i \in \omega_k} b_k - \sum_{i \in \omega_k} I_i^S = 0$$

Dividing by $|\omega_k|$, the number of pixels in the window, gives the means μ_k^E and μ_k^S. Hence, solving for b_k:

$$a_k \mu_k^E + b_k - \mu_k^S = 0$$
$$\implies b_k = \mu_k^S - a_k \mu_k^E.$$

Derivation of the Scaling Coefficient a_k Next, we differentiate E with respect to a_k and set the result to zero:

$$\frac{\partial E}{\partial a_k} = 2 \sum_{i \in \omega_k} (a_k I_i^E + b_k - I_i^S) I_i^E = 0$$

Substituting our expression for $b_k = \mu_k^S - a_k \mu_k^E$:

$$\sum_{i \in \omega_k} (a_k I_i^E + (\mu_k^S - a_k \mu_k^E) - I_i^S) I_i^E = 0$$
$$\implies \sum_{i \in \omega_k} (a_k (I_i^E - \mu_k^E) - (I_i^S - \mu_k^S)) I_i^E = 0$$

Solving for a_k:

$$a_k = \frac{\sum_{i \in \omega_k} (I_i^S - \mu_k^S) I_i^E}{\sum_{i \in \omega_k} (I_i^E - \mu_k^E) I_i^E}$$

This expression is equivalent to the covariance of I^E and I^S divided by the variance of I^E. Dividing the numerator and denominator by $|\omega_k|$ and including the regularization term ρ, we arrive at the final form:

$$a_k = \frac{\frac{1}{|\omega_k|} \sum_{i \in \omega_k} I_i^E I_i^S - \mu_k^E \mu_k^S}{(\sigma_k^E)^2 + \rho}, \tag{14}$$

where $(\sigma_k^E)^2$ is the variance of I^E in ω_k.

D. Algorithms

We provide the overall algorithm of the optimization-driven denoising process in Sec. 3.2 in Algorithm 2. We provide the cross-attention mask upsampling algorithm of Sec. 3.3 in Algorithm 2.

E. Source and Edit Prompt Generation for Image-Based Editing Tasks

Unlike other image editing tasks where a text prompt is provided or can be easily specified, image harmonization and photorealistic style transfer depend on an additional input image that visually encodes the editing instructions. This creates a challenge for text-based editing models, which require these visual instructions to be converted into text prompts. To address this, we employ a multi-modal large language model, such as GPT-4o [43], drawing inspiration from the prompt generation approach in Diff-Harmonization [8].

Image Harmonization. In image harmonization, our aim is to seamlessly integrate a foreground object into a background. We begin by using the mask image to distinguish the foreground and background regions. Next, a vision-language model generates text descriptions for the foreground object (FO), its foreground description (FD), and the background description (BD). Using these, we construct the source prompt ("FD FO") and edit prompt ("BD FO"). The specific template for this is illustrated in Fig. 16, where we feed this prompt into GPT-4 to produce the source and edit prompts.

Photorealistic Style Transfer. In photorealistic style transfer, the goal is to apply the visual style of one image to the content of another. We start by using a vision-language model to create text descriptions for both the content image and the style image. From the style image's description, we extract terms that capture its visual style. Then, we modify the content image's description by replacing its style-related terms with those derived from the style image. Following the template in Fig. 17, GPT-4o generates a source and edit text prompt pair based on this approach.

F. Detailed Related Works on Task-Specific Structure-Preserving Image Editing

We provide additional details on long-standing image editing tasks where it is crucial to preserve the pixel-level structure of the input image. We also briefly discuss the approaches for these tasks that were introduced before diffusion-based image editing. While the results produced by these earlier methods show high structural fidelity due to the specific assumptions they make, this specialization restricts their use in broader editing scenarios.

Image Relighting modifies the illumination of an input image. Recent learning-based approaches primarily rely on physics-based priors and image datasets obtained from a light-stage [12, 66] to achieve realistic relighting results while preserving the underlying scene [28, 42, 44, 58, 77]. Despite their effectiveness, these methods remain specialized for physics-based relighting tasks.

Image Tone Adjustment alters tonal properties—such as brightness, contrast, and color balance—of the input image while preserving its structure. Techniques range from color transformation matrix-based methods [7, 15, 70] to look-up-table-based methods [17, 27, 31, 64, 73]. These methods effectively constrain edits to color space transformations and thereby preserve edges and spatial structure. However, their reliance on fixed transformations limits adaptability to more complex tasks.

Image Harmonization and *Background Replacement* aim to make a composite image visually coherent by adjusting the foreground to match the color statistics and illumination of the background. Traditionally, image gradient-based methods [23, 47, 59, 61] and image color statistics-based methods [10, 48, 52, 68] were proposed, followed by data-driven methods using neural networks [9, 38]. While these methods mostly preserve the input images' structure, they focus narrowly on compositing scenarios.

Photorealistic Style Transfer transfers the reference style onto the input image while preserving style-independent features of the input image. Early works operated similarly to image harmonization by matching the image statistics of the input and reference images [48, 52]. Modern approaches match the statistics of deep features [13, 34], where these features are obtained by feeding the image into a pretrained image classification neural network [57]. Follow-up works have addressed the structural artifacts produced during style transfer, aiming to retain fine structural details of the input image [9, 36, 39, 71]. However, these methods do not account for other types of attribute manipulation other than the overall image style.

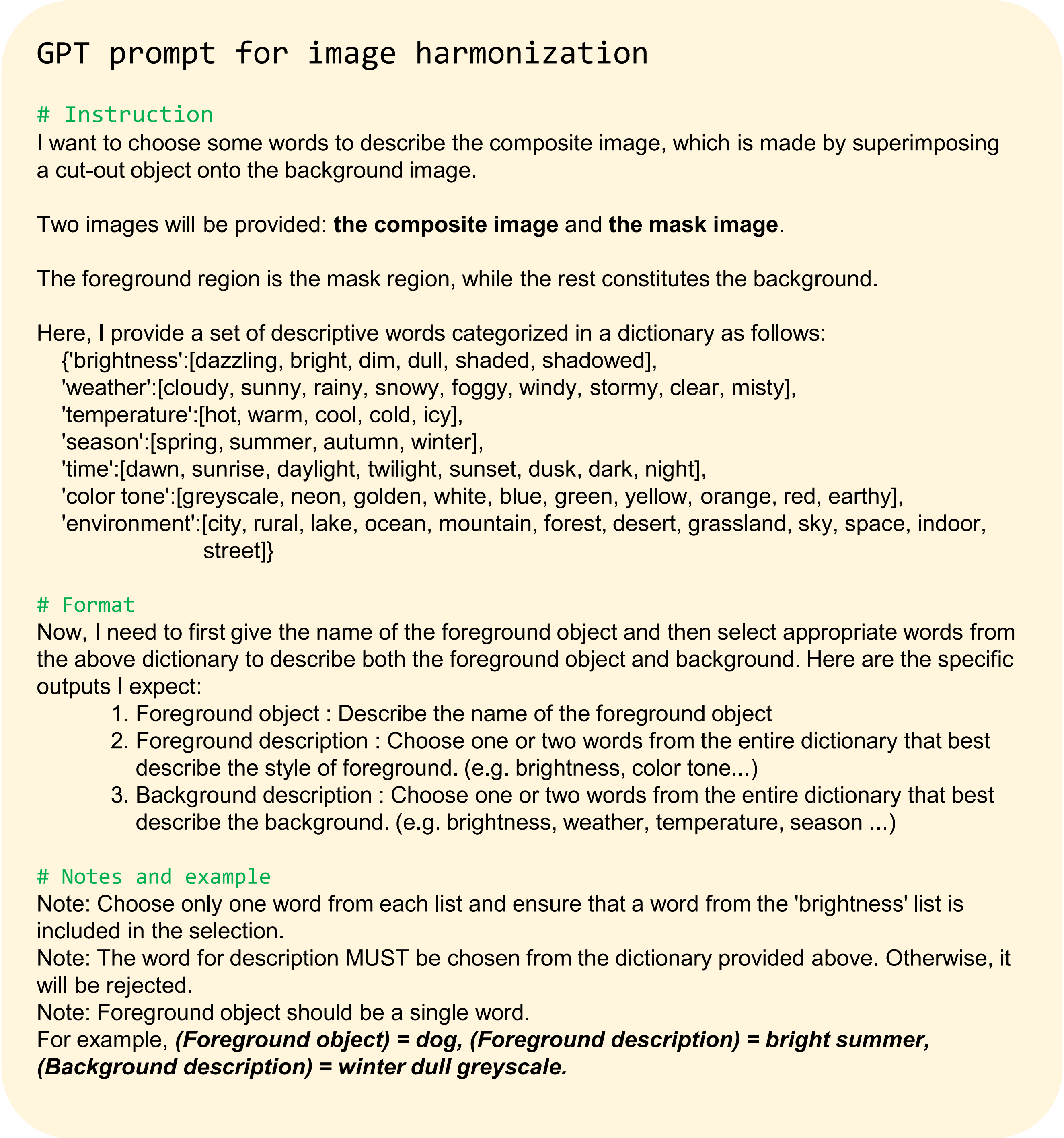
Time-Lapse and *Season or Weather Change* involve hallucinating how a scene would appear with different transient attributes, such as time or weather. Data-driven algorithms have suggested example-based appearance transfer [29, 56], but editing is constrained to domain-specific datasets. Since modifying transient attributes often requires generating new details, GAN-based models have also been introduced [1, 22, 78]. However, they also rely on domain-specific datasets, limiting their application to a particular setting.

Algorithm 1 Optimization-Driven Denoising Process

```
Require: Source image $I_{\text{src}}$, edit prompt $p_{\text{edit}}$, source features $f_t^{\text{src}}$, noise prediction model $\epsilon_\theta$, encoder $\mathcal{E}$, decoder $\mathcal{D}$, maximum timestep $T$, coefficients $a_t, b_t$ from noise schedule, scheduling timestep threshold $t_s$, learning rates $\eta, w$, number of optimization steps $k, s$, loss weights $\lambda$
Ensure: Edited image $\hat{I}_0$
Initialize latent $z_T \sim \mathcal{N}(0, \mathbf{I})$
for $t$ in $(T, 1)$ do
    $\hat{\epsilon}_t \leftarrow \epsilon_\theta\left(z_t, t, p_{\text{edit}}, f_t^{\text{src}}\right)$
    $\hat{z}_0^{(t)} \leftarrow \frac{1}{a_t}\left(z_t - b_t \hat{\epsilon}_t\right)$
    if $t \leq t_s$ then                                        ▷ Scheduling optimization to preserve details
        $\hat{I} \leftarrow \mathcal{D}(\hat{z}_0^{(t)})$                                          ▷ Decode to image space
        for $i$ in $(0, k)$ do                                               ▷ Gradient descent step
            $\hat{I} \leftarrow \hat{I} - \eta \nabla_{\hat{I}} \left\{\mathcal{L}_{\text{SPL}}(I_{\text{src}}, \hat{I}) + \lambda \mathcal{L}_{\text{CPL}}(I_{\text{src}}, \hat{I})\right\}$
        end for
        $\tilde{z}_0^{(t)} \leftarrow \mathcal{E}(\hat{I})$                                       ▷ Re-encode optimized image
    end if
    $z_{t-1} \leftarrow \mathcal{S}\left(\tilde{z}_0^{(t)}, z_t, t, \hat{\epsilon}_t\right)$
end for
$\hat{I}_0 \leftarrow \mathcal{D}(z_0)$
for $i$ in $(0, s)$ do                                          ▷ Post processing in image space
    $\hat{I}_0 \leftarrow \hat{I}_0 - \eta \nabla_{\hat{I}_0} \left\{\mathcal{L}_{\text{SPL}}(I_{\text{src}}, \hat{I}_0) + \lambda \mathcal{L}_{\text{CPL}}(I_{\text{src}}, \hat{I}_0)\right\}$
end for
```

Algorithm 2 Iterative Guided Mask Upsampling

```
Require: Initial cross-attention map $M_{init}$, reference image $I$, target size $T$, initial radius $r$, radius increment $\Delta r$
Ensure: Refined mask $M$
$M \leftarrow$ Binarize$(M_{init}, 0.4)$
$s \leftarrow$ size$(M)$                                              ▷ Get initial resolution.
while $s < T$ do
    $s \leftarrow 2 \times s$
    $M \leftarrow$ BilinearUpsample$(M, \text{scale} = 2)$
    $I_s \leftarrow$ Resize$(I, s)$                          ▷ Downscale reference image to $s \times s$ resolution.
    $M \leftarrow$ GuidedFilter$(M, I_s, r, \epsilon)$
    $r \leftarrow r + \Delta r$
end while
return $M$
```



GPT prompt for image harmonization

# Instruction
I want to choose some words to describe the composite image, which is made by superimposing a cut-out object onto the background image.

Two images will be provided: **the composite image** and **the mask image**.

The foreground region is the mask region, while the rest constitutes the background.

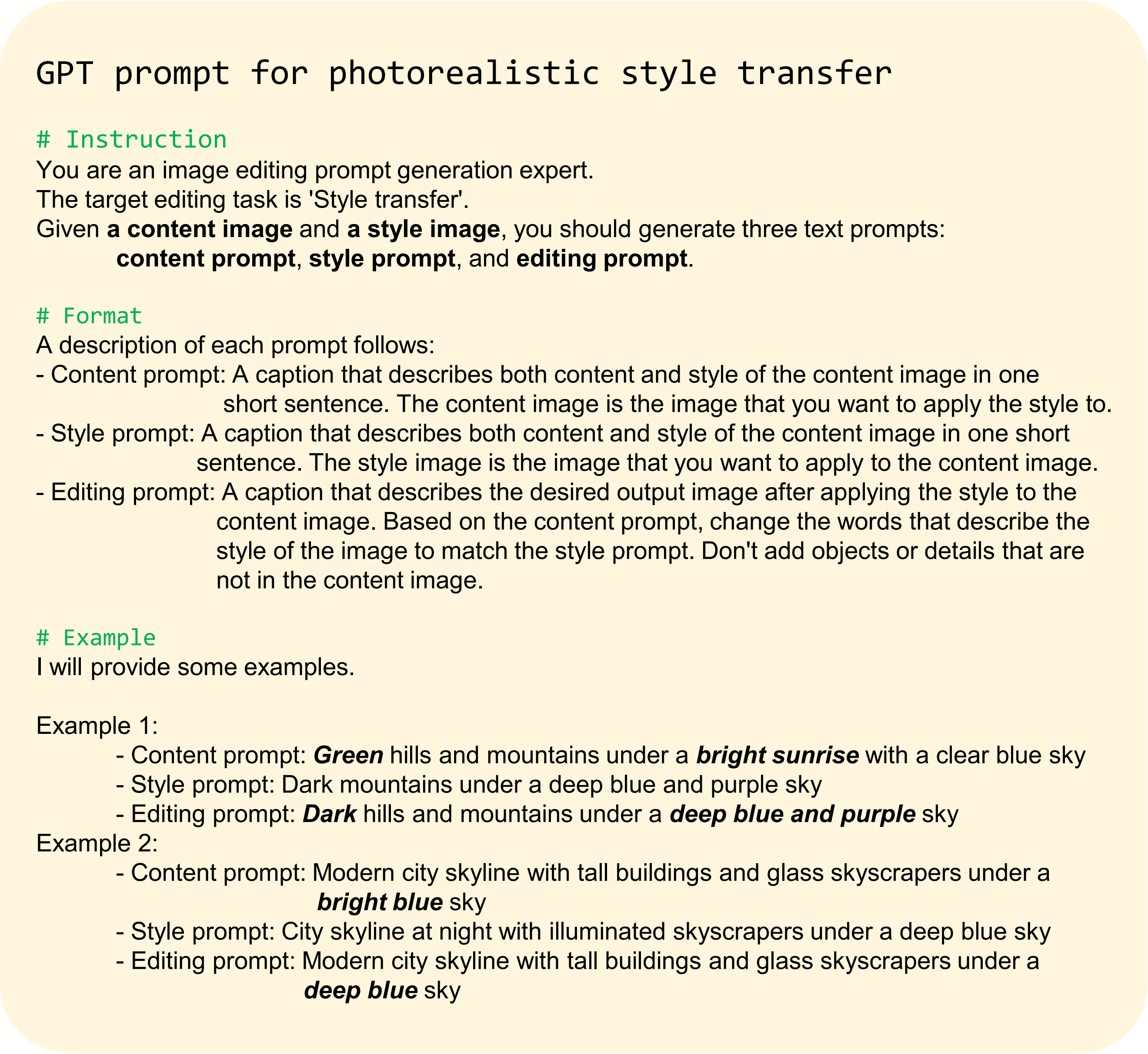
Here, I provide a set of descriptive words categorized in a dictionary as follows:
{'brightness':[dazzling, bright, dim, dull, shaded, shadowed],
'weather':[cloudy, sunny, rainy, snowy, foggy, windy, stormy, clear, misty],
'temperature':[hot, warm, cool, cold, icy],
'season':[spring, summer, autumn, winter],
'time':[dawn, sunrise, daylight, twilight, sunset, dusk, dark, night],
'color tone':[greyscale, neon, golden, white, blue, green, yellow, orange, red, earthy],
'environment':[city, rural, lake, ocean, mountain, forest, desert, grassland, sky, space, indoor, street]}

# Format
Now, I need to first give the name of the foreground object and then select appropriate words from the above dictionary to describe both the foreground object and background. Here are the specific outputs I expect:

1. Foreground object : Describe the name of the foreground object
2. Foreground description : Choose one or two words from the entire dictionary that best describe the style of foreground. (e.g. brightness, color tone...)
3. Background description : Choose one or two words from the entire dictionary that best describe the background. (e.g. brightness, weather, temperature, season ...)

# Notes and example
Note: Choose only one word from each list and ensure that a word from the 'brightness' list is included in the selection.
Note: The word for description MUST be chosen from the dictionary provided above. Otherwise, it will be rejected.
Note: Foreground object should be a single word.
For example, ***(Foreground object) = dog, (Foreground description) = bright summer, (Background description) = winter dull greyscale.***

Figure 16. A prompt for using GPT-4o [43] as a prompt generator for image harmonization.



GPT prompt for photorealistic style transfer

# Instruction

You are an image editing prompt generation expert.
The target editing task is 'Style transfer'.
Given **a content image** and **a style image**, you should generate three text prompts:
content prompt, **style prompt**, and **editing prompt**.

# Format

A description of each prompt follows:
- Content prompt: A caption that describes both content and style of the content image in one short sentence. The content image is the image that you want to apply the style to.
- Style prompt: A caption that describes both content and style of the content image in one short sentence. The style image is the image that you want to apply to the content image.
- Editing prompt: A caption that describes the desired output image after applying the style to the content image. Based on the content prompt, change the words that describe the style of the image to match the style prompt. Don't add objects or details that are not in the content image.

# Example

I will provide some examples.

Example 1:
- Content prompt: ***Green*** hills and mountains under a ***bright sunrise*** with a clear blue sky
- Style prompt: Dark mountains under a deep blue and purple sky
- Editing prompt: ***Dark*** hills and mountains under a ***deep blue and purple*** sky

Example 2:
- Content prompt: Modern city skyline with tall buildings and glass skyscrapers under a ***bright blue*** sky
- Style prompt: City skyline at night with illuminated skyscrapers under a deep blue sky
- Editing prompt: Modern city skyline with tall buildings and glass skyscrapers under a ***deep blue*** sky

Figure 17. A prompt for using GPT-4o [43] as a prompt generator for photorealistic style transfer.