Zero-Shot Depth Aware Image Editing with Diffusion Models

Supplementary Material

Appendix

Contents

Algorithm 1 Algorithm of proposed approach

Scene Compositing
Define: ω_{mdf}: guidance strengths; N: number of guidance steps; T_c: annealed timestep; I_{fg}, I_{bg}: foreground, background (fg/bg) images; d_{fg}, d_{bg}: fg,bg depths; c_{fg}, c_{bg}: fg,bg text prompts; C_c: composed scene text prompt; θ_{sd}: depth conditioned SD-v2; d_p: depth value for composing the scene Ψ_t: diffusion features at timestep t
$(x_T^{fg}, \Psi_{1:T}^{fg}) \leftarrow \text{NullInversion}(\theta_{sd}, I_{fg}, c_{fg}, d_{fg})$
$(x_T^{bg}, \Psi_{1:T}^{bg}) \leftarrow \text{NullInversion}(\theta_{sd}, I_{bg}, c_{bg}, d_{bg})$
$M_{fg},\ M_{bg} \leftarrow \text{DeGLaD}(d_{fg},\ d_p)$ ▷ Getting layered masks
$d_c \leftarrow d_{fg} \otimes M_{fg} + d_{bg} \otimes (1 - M_{fg})$
initialize: $x_T \leftarrow x_T^{bg}$
for $t = T$ **to** 1 **do** ▷ denoising loop
  if $t > T_c$ **then** ▷ apply guidance only for $t < T_c$
    for $n = 1$ **to** N **do** ▷ guidance loop
      $\Psi_t^{edit} \leftarrow extractfeatures(\theta_{sd}, x_t,\ C_c,\ d_c)$
      $x_t \leftarrow x_t - w_{mdf} \nabla_{x_t} \mathcal{L}(\Psi_t^{fg}, \Psi_t^{bg}, \Psi_t^{edit})$ ▷ guid. step
    end for
  end if
  $x_{t-1} \leftarrow Updatestep(x_t, \theta_{sd})$ ▷ single denoising step
end for
Object Placement
Define: ω_{mdf}: guidance strengths; N: num. guidance steps; T_c: annealed timestep; I: input background image; d: input image depth; c: input image text prompts; θ_{inp}: AnyDoor (inpainting model); c_{inp}: input box/mask for inpainting; d_p: depth value for placing object
$(\Psi_t,\ x_T) \leftarrow \text{NullInversion}(\theta_a, I, c_{inp})$
$M_{fg},\ M_{bg} \leftarrow \text{MPI}(d,\ d_p)$
initialize $x_T \leftarrow \mathcal{N}(\mu,\ \sigma^2)$
for $t = T$ **to** 1 **do** ▷ denoising loop
  if $T_c > t > 0$ **then**
    for $n = 1$ **to** N **do** ▷ guidance loop
      $\Psi_t^{edit} \leftarrow extractfeatures(\theta_{inp}, x_t,\ c_{inp})$
      $x_t \leftarrow x_t - w_{mdf} \nabla_{x_t} \mathcal{L}(\Psi_t, \Psi_t^{edit}))$ ▷ guidance step
    end for
  end if
  $x_{t-1} \leftarrow Updatestep(x_t, \theta_{inp}, c_{inp})$ ▷ single denoising step
end for

A. Project Page and Algorithm

Please check the https://rishubhpar.github.io/DAEdit/ for high-resolution visual results. We have provided the detailed algorithm for both of our depth-aware editing tasks - scene composition and object insertion in Alg.1.

B. User interface for providing depth input

Our method requires an input depth value d_0 for object placement and composing scenes at a particular depth. Providing depth value for these tasks can be challenging to the user given a single image, and the user may have to do multiple trials to obtain the desired depth value. To this end, we create a simple user interface to easily obtain the depth value for editing. We first apply the SAM [9] image segmentation model on the background scene and lift the segmentation map in 3D as point clouds using the input depth map. Next, we visualize the segmented point cloud from the Bird's Eye View (BEV) as shown in Fig. 1. The BEV projection is a simpler representation and makes it easier to understand the placement of the scene objects. A user can select a point in BEV projection where they want to place the object, and the corresponding depth value will be used for placement. Alternatively, the above process can be automated to accept input placement prompts (*'An object between **sofa** and the **wall**'*) by selecting a depth value

between the average depth of the sofa and the wall.

C. Ablations

We present ablations over crucial hyperparameters used in the proposed approach.

C.1. Layered latent α - Comp vs `FeatGLaC`.

We ablate over the choice of the feature space to implement layered representation for effective *depth-aware* compositions. The ablation is provided in Fig. 2a). We apply layered latent α-compositing at some intermediate denoising timestep τ to compose the foreground and background scenes. For the remaining $T - \tau$ timestep, we let the composed latent denoise freely to generate a natural-looking composite image. However, applying Layered α-compositing in the latent space at τ suffers from a tradeoff and leads to inferior scene composition results. Having a higher τ value does not allow enough flexibility for the edited latent to blend both regions. On the contrary, having a lower τ value results in realistic blending but has a significant loss in the identity of the foreground and the background region. Our guidance method - `FeatGLaC` addresses this issue and achieves realistic blending while preserving foreground and background regions.

C.2. Layered latent guidance vs `FeatGLaC`.

We perform an ablation over applying the guidance in layered latent space vs layered representation in U-Net feature space in Fig. 2b). Guidance in layered latent space has a tradeoff in the output generation, where applying guidance for a small number of timesteps results in significant identity change for the input scene, and guidance for more steps results in unnatural blending in the output. In contrast, applying the guidance in the more expressive U-Net feature representation results in natural scene compositing (improved illumination effect on the foreground) along with identity preservation.

C.3. Layered feature space editing vs Layered feature space guidance.

We also compare with a direct fusion of layered U-Net features of the foreground and background image in Fig. 2c). Manipulating the U-Net features is fragile and results in an unnatural composition compared to manipulating the diffusion latents. Our method uses the layers obtained from `DeGLaD` in U-Net features to guide the diffusion latent slowly, which is the most effective and keeps the diffusion latent in the distribution of the original distribution, resulting in natural results.

C.4. Number of optimization steps.

We ablate over the number of optimization steps for guiding the latent at each step of denoising in Fig. 3. Using a small number of optimization steps results in significant identity loss of the scene, and using a large number of optimization steps can lead to artifacts in the scene's appearance as it can significantly alter the latent distribution. We use 5 optimization steps for scene composition and 3 optimization steps for object insertion, as the underlying inpainting model in object insertion inherently preserves the scene background.

C.5. Guidance timestep for Object placement.

We ablate over the number of timesteps used for guidance for object placement in Fig. 5. For object placement, we first perform `DeGLaD` to get a layered representation, which is used for latent α-compositing at intermediate timestep τ, and then feature guidance is used for the remaining $T - \tau$ timesteps. Performing the fusion at later timesteps ($T = 40$) results in a 'cut-paste' appearance, resulting in unnatural compositing, and the object can get cropped due to an inaccurate mask. On the contrary, fusing early ($T = 20$) results in a strong prior from the inpainting model, which generates the object at with an inaccurate occlusion.

C.6. Ablation over depth model.

Our method requires a depth map to obtain the layered representation from `DeGLaD` for scene editing. We ablate over different depth models to evaluate the robustness of our framework. We predict depth maps from Marigold [7], Midas [11] and Depth-Anything [15] model and perform object placement in Fig. 4.

C.7. Hyperparameter ablation

We conduct a quantitative ablation study on key hyperparameters, including the number of timesteps used for guidance and the weight assigned to the guidance loss. The results are presented in Tab. 1 and Tab. 2, respectively. To evaluate the quality of generated images, we use LPIPS to assess how well the appearance is preserved and KID (computed on the COCO training dataset) to measure the realism of the generated images. Additionally, we employ Image Reward [13] metrics, a model trained to predict human preference for an image given a specific prompt. This metric evaluates how well the generated images align with the prompt and estimates a human-assigned score, and we see that the hyperparameter we chose visually also gives us the best quantitative metrics.

D. Dataset details

D.1. Scene compositing.

We collect background images from the SSharmonization dataset [5], and for foreground images, we take a variety

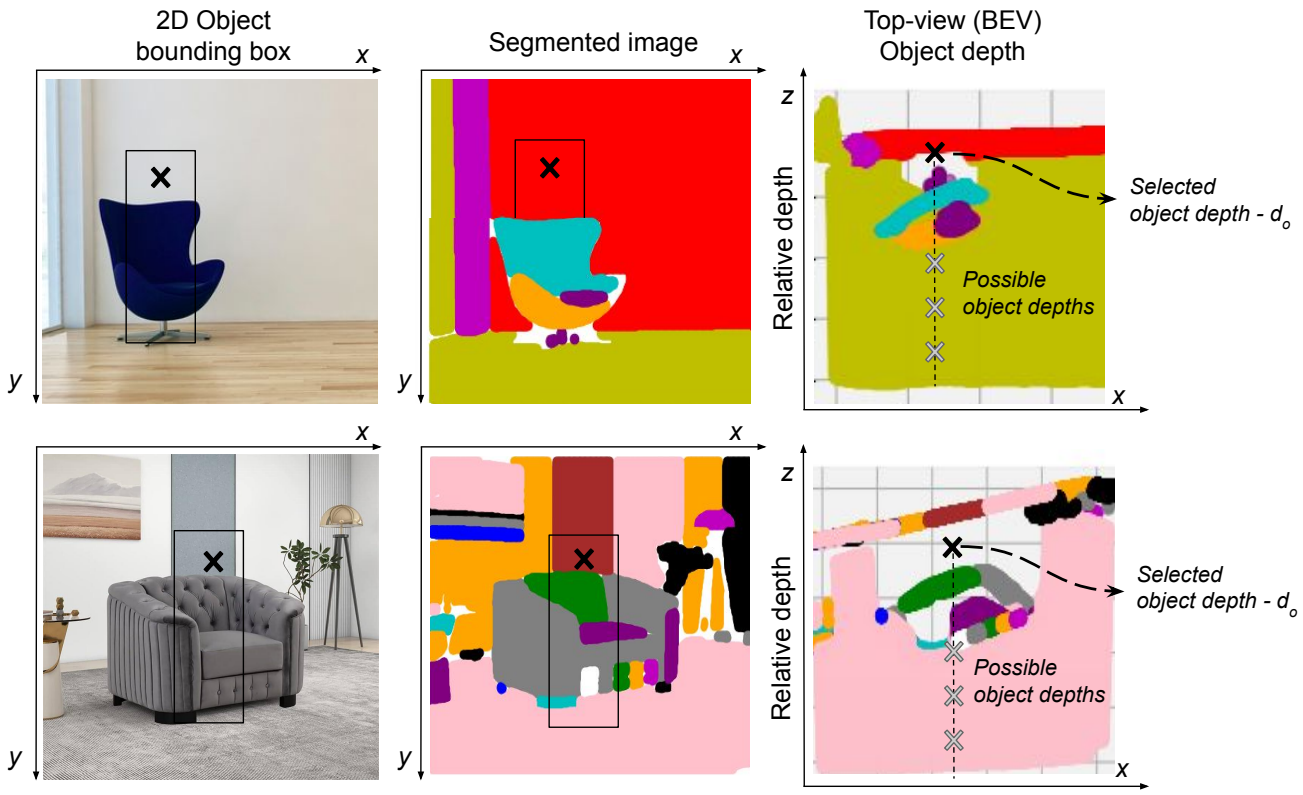

Figure 1. **User interface to provide object depth as input.** Given a background image with its depth map and a 2D bounding box, we segment the scene and lift the segmentation map to a 3D point cloud using the provided depth map. Next, we visualize the segmented point cloud from Birds' Eye View (BEV) for selecting the object depth. The BEV representation provides a convenient visualization for selecting object depth, where the user can just select a point, and the corresponding depth value d_o will be used to place the object.

Table 1. Ablation over guidance timestep

Timestep	LPIPS ↓	KID x 10-2 ↓	IR ↓
10	0.552	5.6	-0.557
20	0.416	**4.8**	0.065
30	0.305	4.9	0.72
38	0.247	5.4	**0.836**
40	0.238	5.4	0.833
49	**0.231**	5.4	0.758

Table 2. Ablation over guidance weight

Weightage	LPIPS ↓	KID x 10-2 ↓	IR ↓
0.3	0.369	**5.1**	-0.27
0.6	0.272	5.3	0.73
1.0	0.247	5.4	**0.836**
1.2	**0.245**	5.3	0.807
1.5	0.251	5.4	0.819

of images with different lighting from Google images. Our dataset consists of around 2844 images with 80 background images and 36 foreground images. To get the foreground mask, we manually do an annotation to find the best MPI plane where we can get a meaningful foreground region that can be composed with other background images. To get the prompt for the background and foreground scene, we use an off-the-shelf image captioning model [1] to obtain the text prompts. And to get the combined scene prompt, we use simple heuristics like *'a photo of a {foreground object} with {background scene prompt} in the background'*.

D.2. Object Placement.

Our object placement evaluation dataset consists of 491 background-object image pairs. All of these are collected from Google images, consisting of outdoor and indoor scenes with high diversity in illumination and appearances. We manually annotate the background image with a plausible object bounding box and a depth value where the object can be placed meaningfully with proper occlusion. A user can use a simple interface (discussed in Sec.2 and Fig. 1) in the birds' eye view of the scene to define the appropriate depth for object placement. Here also, we use [1] for annotating the background scene, which is needed for null text inversion.

Figure 2. a) **Ablation with `DeGLaD` in latent representation.** Fusing `DeGLaD` latents at the initial timestep leads to a significant loss in the identity of the input scene, and fusion at a later timestep results in unnatural blending. Our `FeatGLaC` generates a realistic composite image while preserving the identity of the background scene. b) **Ablation with layered latent guidance.** Instead of guiding the layered feature, we apply the guidance on the layered latent representation. Guidance on latents does not have a significant change from the layered latent fusion in a) and results in unnatural results. **c) `DeGLaD` fusion in U-Net feature space.** Applying `DeGLaD` fusion directly at U-Net features instead of using layered feature guidance leads to significant changes in the latents, resulting in unnatural compositions. Our approach to slowly update the latents with guidance keeps the latents in distribution and results in realistic compositing.

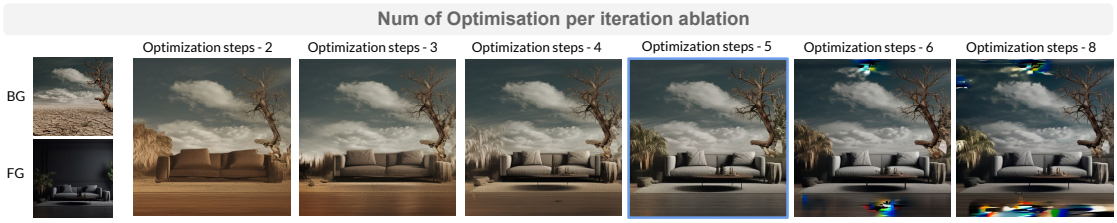

Figure 3. **Ablation over the number of optimization timesteps.** Using a small number of optimization steps results in significant identity loss of the scene, and using a large number of optimization steps can lead to artifacts in the scene's appearance as it can significantly alter the latent distribution. We use 5 optimization steps.

Figure 4. **Ablation over depth predictors for object placement.** Our approach is robust to the choice of monocular depth predictor and results in consistent edits.

E. Implementation details

We perform ablation using different layers for guidance loss calculation, and we observe that using the last two layers of the u-Net seems to be effective for both of our applications. For the task of scene composition, instead of starting from random noise, we start from the DDIM inversion of the background image and then apply guidance from both the background and the foreground regions. Starting from the background layer, the latent results in the scene lighting are inherited from the background scene. This is particularly important in relighting applications, where we want to illuminate the foreground region with the lighting from the background. We give guidance from 0th timestep to 38, similar to giving guidance till $50th$ timestep, causes it to look similar to cut and paste without any scene effects such as illumination. We use 5 optimization steps per iteration for scene compositions and 3 steps for object placement. We use less number of steps for object placement since the inpainting model already does well in preserving the appearance of the features outside of the bounding box. Specifically, for object placement, we perform latent α-comp at intermediate timestep τ with the layered mask we obtain from `DeGLaD` and then perform feature guidance for the rest of the timesteps to preserve the foreground. There is a tradeoff between which timestep τ to use, and we found $\tau = 30$ works best in most of the cases. The time taken to generate a single image for object placement is 60 sec, and for scene composition, it takes around 86 sec.

F. Additional Results

F.1. Additional Object Insertion Comparisons

We provide an additional comparison for object insertion with a layered representation baseline, where we apply `DeGLaD` in image space to obtain a layered representation and then place the object at the desired depth. However, the inserted object does not blend well with the background scene, as the layered representation in the image space does not affect the color of the object placed. To this end, we apply image harmonization [8] on top of the image space α-comp output to generate realistic object insertion. We present our results in Fig. 10, along with other inpainting based approaches like IP-Adapter [16], Paint by example [14], Anydoor [4] and few recent object inpainting methods such as Brushnet [6], Mimic Brush [3], and DiptychPrompting [12].

Method	DINO-sim ↑	KID ↓	Δ depth ↓	Clip-sim ↑
IP-Adapter [16]	0.244	5.3	9.366	27.81
DeGlad+Harmonization	**0.576**	**4.7**	2.985	68.5
LoMOE []	0.206	5.1	6.49	48.46
Brushnet [6]	0.209	5.0	8.04	43.96
PbE [14]	0.273	4.9	6.733	60.12
Mimic Brush [3]	0.538	4.9	5.76	76.68
DiptychPrompting [12]	0.260	4.8	6.30	42.33
Anydoor [4]	0.507	4.9	3.176	83.23
Ours	0.545	4.8	**2.989**	**84.86**

Table 3. **Depth-Aware object insertion additional comparison.** KID and Δ **depth** are reported in x10^2 units.

F.2. Object Insertion + MasaCtrl [2]

Our object placement framework uses a pre-trained object inpainting diffusion model [4] as the backbone. Hence, the identity of the generated object is limited by the identity preservation capability of the base inpainting network. However, we can improve upon the object identity by performing a refinement step. Specifically, we use MasaCtrl [2], with the reference object image and the edited image. MasaCtrl injects the identity features in the generated object region, resulting in improved identity. We present the result in Fig. 7, where applying MasaCtrl improved the object appearance and recovered some of the fine features from the vase.

F.3. Object Insertion without Inpainting model

Object insertion can be performed without relying on an inpainting model; however, using an inpainting model offers additional advantages such as adjusting the object's orientation and generating realistic shadows. For object insertion without inpainting, we can adopt a strategy similar to scene composition. Specifically, we use a depth-conditioned diffusion model to blend the object into the scene. Instead of a general foreground region, we treat the object mask as the foreground and the remaining scene as the background. The diffusion model then generates a coherent image where the object is naturally composed into the scene according to the provided mask as shown in Fig. 6 and Fig.5c in main paper. However, this approach requires the user to manually and accurately align the object mask to the desired location within the scene.

F.4. Lighting control in Scene Compositing

Our guidance method also allows us to control the lighting change effect that happens during the compositing of two different scenes. In Fig.9, we can see that increasing the guidance weightage reduces the relighting effect and generates a composited image which looks closer to Image cut-

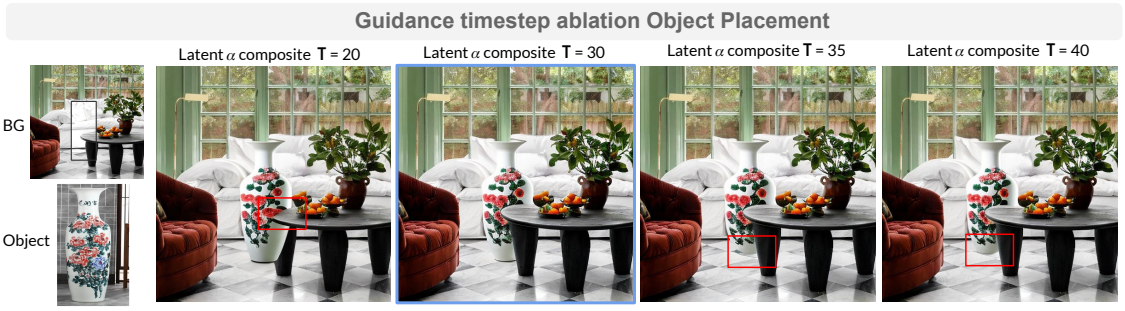

Figure 5. Guidance timestep ablation for object placement. Selecting intermediate timesteps for fusion results in natural scene compositing with accurate occlusions and realistic blending.

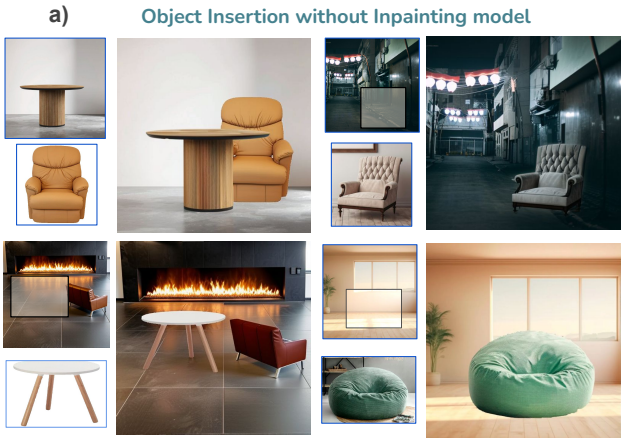

Figure 6. Results for object insertion without using any inpainting methods

Figure 7. Our method when combined with Masa-Ctrl [2], improves identity of the inserted subject.

paste, and lowering the guidance weightage causes the foreground region to follow the background scene lighting.

F.5. Scene compositing comparisons

We provide additional comparison results for scene compositing in Fig. 11. Our method is able to achieve realistic

Figure 8. Qualitative comparison with Object 3DIT [10]

Figure 9. Our Guidance loss weightage can be used to control the strength of relighting, unlike other feedforward-based relighting methods [8].

scene compositing without any large-scale training.

F.6. More results

We present more results for both scene compositing and object placement in Fig. 13 and Fig. 12.

G. User Study

Object placement. We compare our object insertion method against three baselines: IP Adapter, Paint by Example, and Anydoor. The evaluation focuses on three key goals: scene realism, identity replication of the placed object and background, and accurate placement at the intended depth. To assess these goals, we carried out a user study

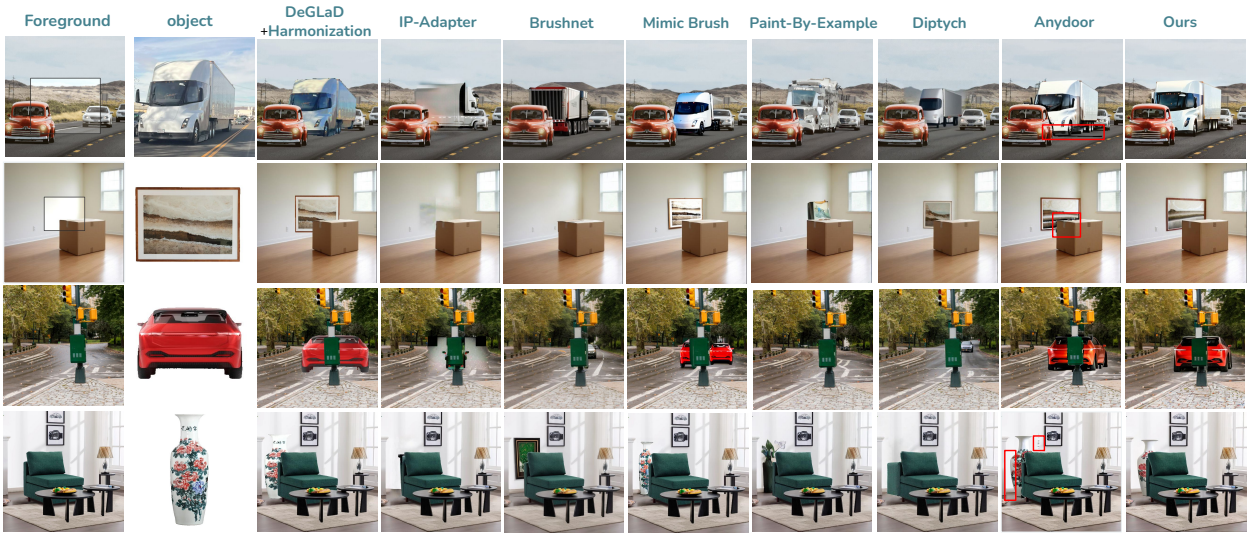

Figure 10. **Comparison of depth-aware object placement:** Image + Harmonization results in an unnatural 'cut-paste' appearance for the inserted object. Inpainting models, IP-adapter, and Paint by example struggle to insert objects with consistent identity given the amodal bounding box. Anydoor achieves decent placement but has significant artifacts at the mask border (marked in red). Our method achieves realistic object placement while preserving the object identity and scene consistency.

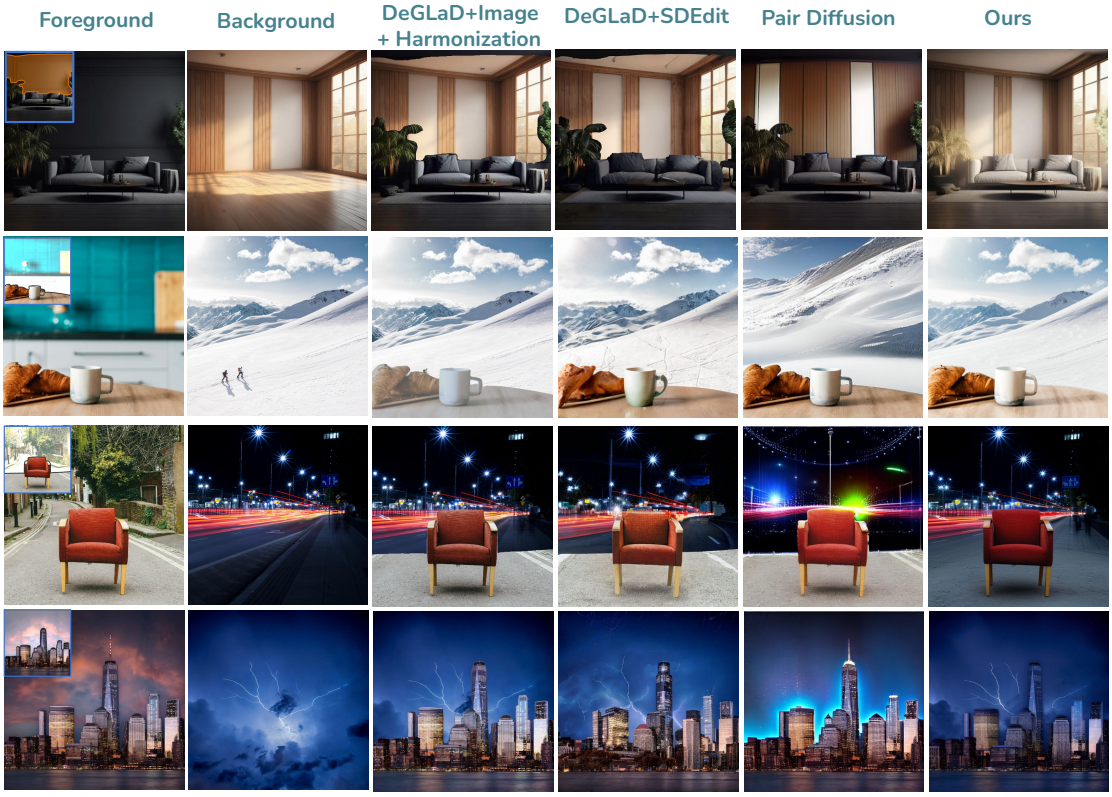

Figure 11. **Comparison for depth-aware scene compositing:** `DeGLaD` + SDEdit and PAIR Diffusion generate unnatural compositions and distort the identity of objects in some cases. Our approach realistically blends the two scenes in a *depth-aware* manner with consistent intra-scene illumination.

on 15 edits across 15 images from our Object Placement dataset. Each goal was evaluated separately by presenting users with pairs of images and asking them to select the one that better met the specific goal. A total of 45 randomized image pairs were generated, with each pair comparing a result from our method to a corresponding result from a ran-

domly chosen baseline. These pairs were divided into three groups of 15 pairs each, corresponding to the three goals. The study involved 40 participants with varied experience in image editing, who evaluated all 15 pairs for each goal, resulting in 600 data points per goal and 1800 in total.

Scene compositing. We compare our approach against `DeGLaD` image baseline + Harmonization, `DeGLaD` + SDEdit, and Pair Diffusion, focusing on two goals: realism and depth consistency. Using a subset of 15 edits across 15 images from our Scene Compositing dataset, we generated 40 image pairs, split evenly across the two goals. The same 40 users participated in this evaluation, generating 800 data points per goal, for a total of 1600 data points.

H. Comparision with 3D editing method

We perform a qualitative comparison with Object3DIT, an object-centric editing model trained on a large-scale, extensively labeled synthetic dataset with 3D annotations. We evaluate their method on the task of object placement and observe that, due to the constraints of their training dataset, it struggles with complex real-world objects and scenes (see Fig. 8). In contrast, our method is training-free, leverages a generic model, and works effectively with diverse objects in real-world scenes.

I. Limitation

Our framework is based on pretrained diffusion models and inherits the limitations and biases of the base model, such as geometrically inconsistent shadows and perspectives in some cases. Further, as we apply guidance at each denoising step, the proposed method is slower than generation from the base model. Our scene compositing doesn't follow actual light transport, but uses diffusion priors to generate a scene which looks plausible in the given scene lighting. For scene relighting, our approach relies on the extent of the background region in the composed image; if the background occupies a small number of pixels, then the relighting will not be effective. This is closer to image harmonization rather than accurate scene relighting. However, our goal is to use depth-based layering and diffusion priors to perform depth-aware edits without any fine-tuning. Using this representation, along with training for a single task, can resolve some of these issues.

References

[1] Abdou. vit-swin-base-224-gpt2-image-captioning. In *https://huggingface.co/Abdou/vit-swin-base-224-gpt2-image-captioning*, 2022. 3

[2] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22560–22570, 2023. 1, 5, 6

[3] Xi Chen, Yutong Feng, Mengting Chen, Yiyang Wang, Shilong Zhang, Yu Liu, Yujun Shen, and Hengshuang Zhao. Zero-shot image editing with reference imitation. *Advances in Neural Information Processing Systems*, 37:84010–84032, 2024. 5

[4] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6593–6602, 2024. 5

[5] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. Ssh: A self-supervised framework for image harmonization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4832–4841, 2021. 2

[6] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 5

[7] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 2

[8] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson W.H. Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *European Conference on Computer Vision (ECCV)*, 2022. 5, 6

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[10] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[11] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 2

[12] Chaehun Shin, Jooyoung Choi, Heeseung Kim, and Sungroh Yoon. Large-scale text-to-image model with inpainting is a zero-shot subject-driven image generator. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7986–7996, 2025. 5

[13] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023. 2

[14] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin

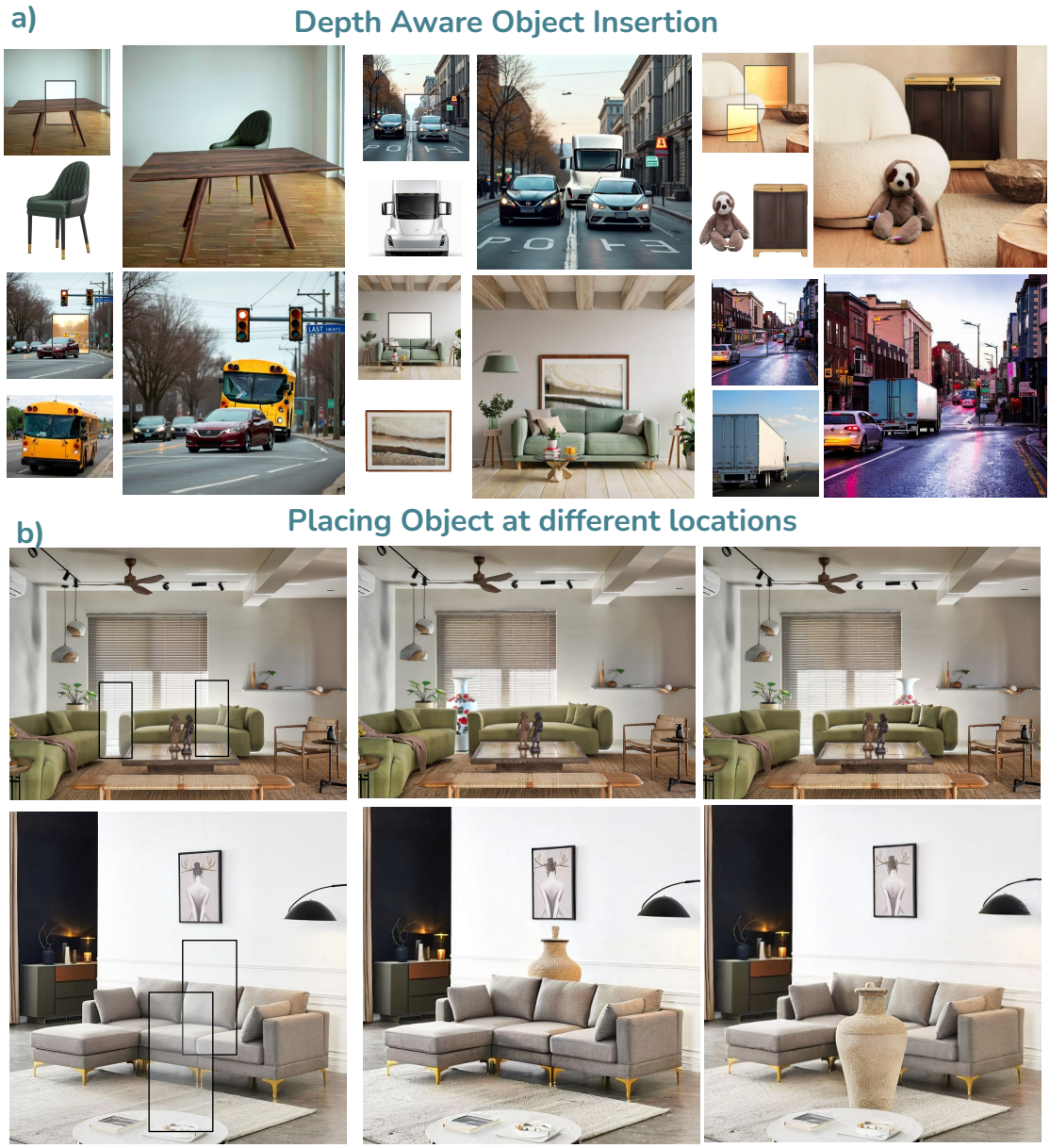

Figure 12. a) Results for depth-aware object placement. b) Our method can also place the given object at multiple locations in a depth-consistent manner

Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 5

[15] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 2

[16] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 5

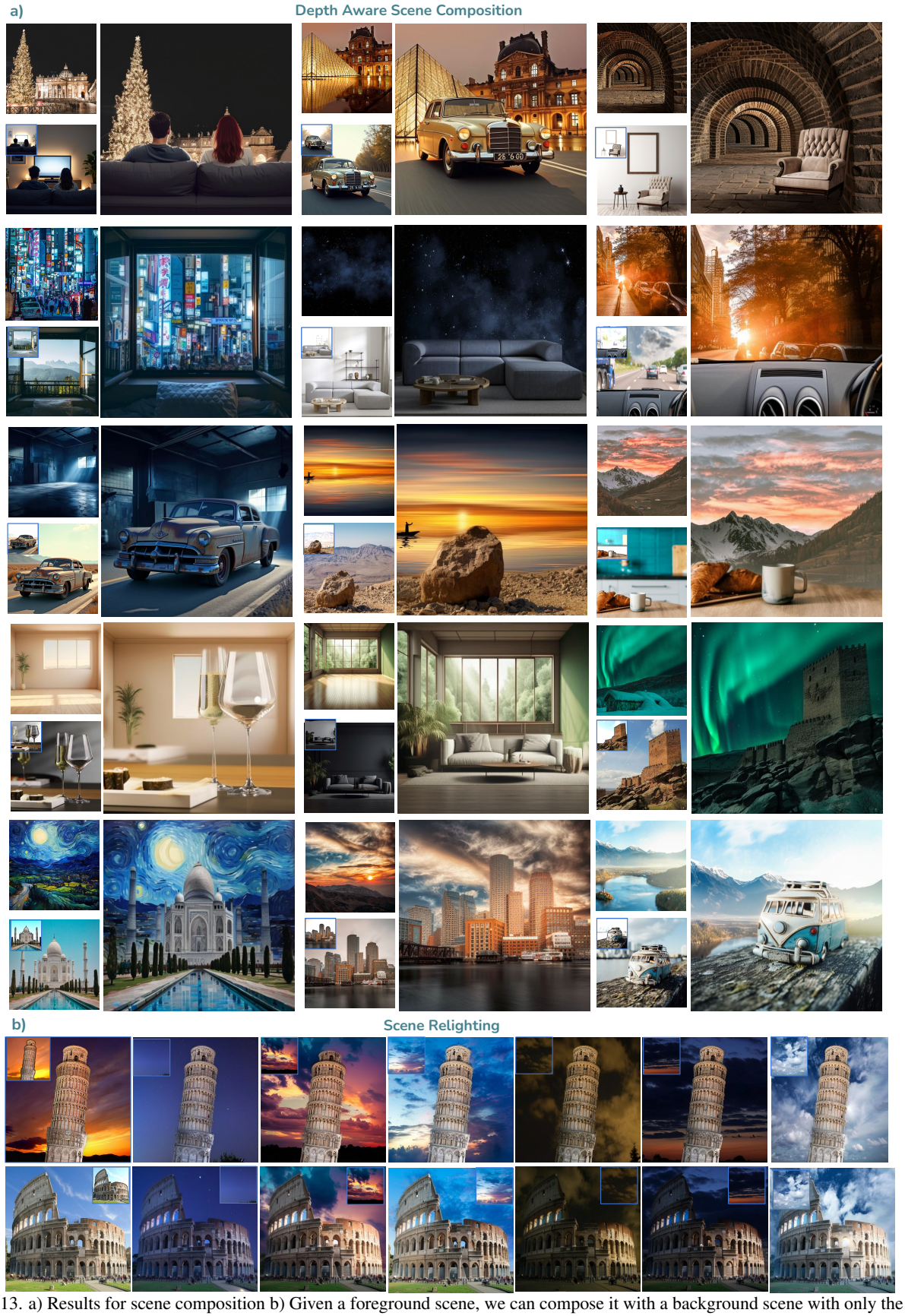

Figure 13. a) Results for scene composition b) Given a foreground scene, we can compose it with a background scene with only the sky to achieve realistic lighting of the foreground subject.

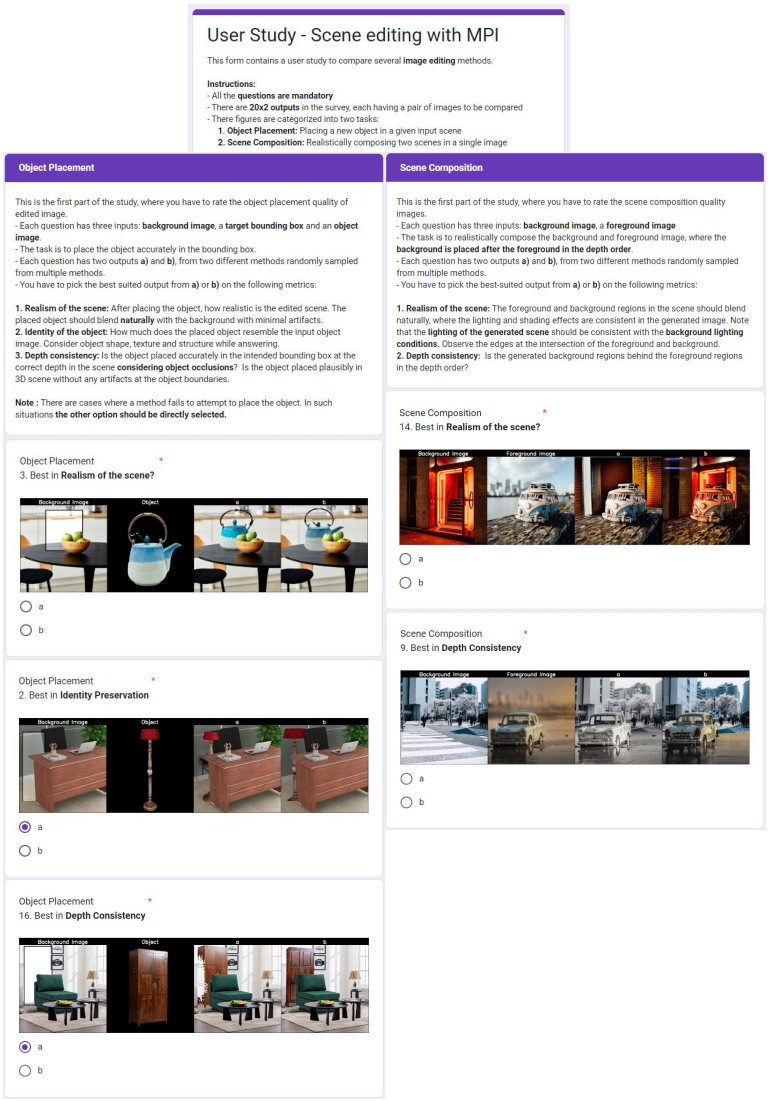

Figure 14. **Sample from user study.** We asked three types of questions on Object Placement (left): Realism of the generated image, Identity of object and background, Depth accuracy and two types of questions on Scene Composition (right): Realism and Depth Accuracy.