DiffDecompose: Layer-Wise Decomposition of Alpha-Composited Images via Diffusion Transformers

Supplementary Material

Overview. In this supplementary material, we first present more discussions about our proposed LDAC task and AlphaBlend dataset. Then, we introduce more information about the training and inference process of DiffDecompose. Subsequently, we offer more competing results upon our proposed AlphaBlend dataset and a public dataset, LOGO, and also include additional quantitative experiments and complexity analyses. Subsequently, we present more visual results from the generative results of DiffDecompose. Finally, we discuss its limitations and broader impacts. The supplement includes the following sections:

A. LDAC Task and AlphaBlend Dataset

LDAC task is no longer equivalent to tasks such as object removal [1, 7, 12] or image matting [5, 6, 9, 11]. The main differences are as follows: (1) The image matting task utilizes Trimap, coarse segmentation mask, and low-quality alpha matte, which the main challenge is to solve the problem of discontinuous edges when removing objects. (2) The object removal task utilizes the mask and prompt to guide the network to recognize its removal region, which the main challenge is how to accurately predict its occlusion region. These two tasks fail to realize our goal: Decomposing an Alpha-composited image into an RGBA foreground image and a RGB background image at the same time. Compared to matting, a prime example is the car window occlusion task in LDAC. Car windows may contain stains or fog, and a mask alone cannot effectively guide the network to cut out the entire image. Furthermore, the object removal task requires complete prediction of the pixels in the removal region, while LDAC separates the removal region from its corresponding background pixel values using conditional probability.

As shown in Figure 1, the construction process of the AlphaBlend dataset is totally different from matting and object removal tasks. The matting dataset is constructed primarily by adjusting the alpha channel values and simply overlaying or linearly superimposing the RGB channel values with background information. This means that Trimap can guide the network to accurately predict the alpha channel to obtain foreground information. However, the AlphaBlend dataset is constructed under non-linear pixel occlusion conditions. A typical example is the contraband scanning scene in security checks, where colors tend to darken. Therefore, the dataset construction process uses a non-linear multiplication of foreground and background for color overlay, which is unrelated to the information provided by Trimap. The object removal dataset is constructed with localized occlusion, and many solid foreground objects cover background pixels. However, when faced with full-image object subtraction scenes from the AlphaBlend dataset, as seen in the ablation experiments section, the full-image mask is completely unable to remove the car window occlusion scene.

It is important to note that the matting task can only subtract local objects. However, the suitcase and cells in this paper are both color overlay tasks, meaning they require not only separating the foreground information but also accurately predicting the corresponding color. For example, a cell immersed in cell sap will turn dark purple, but the actual separated cell will be light purple. This is not a matter of alpha channel weighting but a change in the pixel value itself, which the matting task cannot handle. Furthermore, the object removal task can only recover local image removal; therefore, our experiments only compared the background restoration task of object removal.

B. Training Loss Function

The loss function is formulated within the conditional flow matching [2] framework as the joint expectation over time steps, noisy latent variables, and noise samples:

$$\mathcal{L}_{lce} = \mathbb{E}_{t,p_t(x_t|\epsilon_x),p_t(y_t|\epsilon_y),p(\epsilon_x),p(\epsilon_y)} \left\| v_\Theta(z, t, x, y, c_T) - u_t(z \mid \epsilon_x, \epsilon_y) \right\|^2, \tag{1}$$

where $v_\Theta\,(z, t, x, y)$ are the neural network's predicted conditional velocity fields for the noisy foreground and background latent variables x_t and y_t, respectively, conditioned on the timestep t, the observed composite image z, and conditioning information τ. The true velocities $u_t(z \mid \epsilon_x, \epsilon_y)$, are derived from the diffusion process and noise samples ϵ_x,

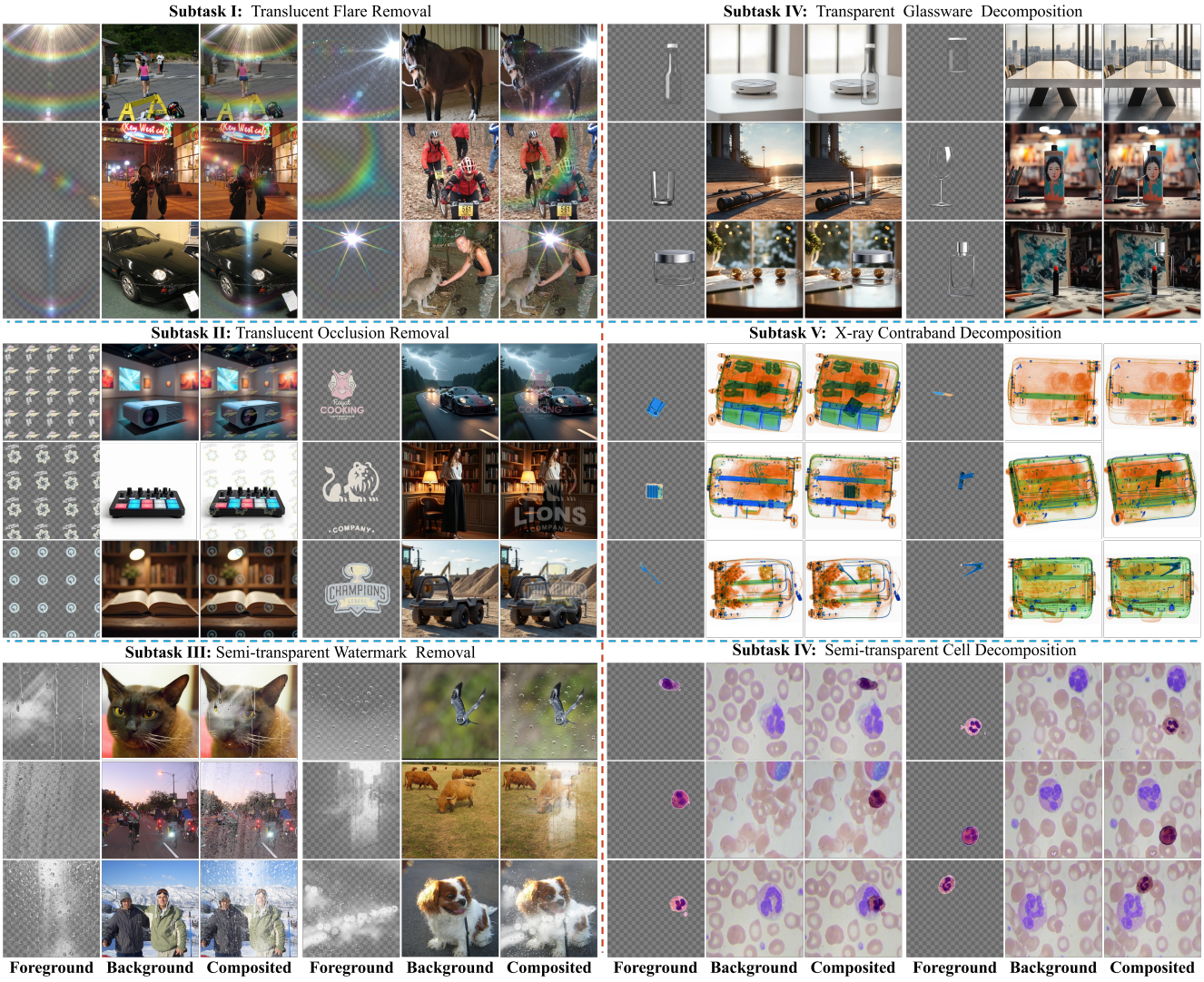

Figure 1. The presentation of the six subtasks' dataset. Each foreground has its respective properties.

ϵ_y. This loss enables the model to learn an optimal probabilistic path from noise to the true decomposed images, thereby jointly inferring foreground and background layers under complex nonlinear composition.

C. Inference Algorithm

Algorithm 1 outlines the inference process of DiffDecompose. In this method, the Alpha-composited image z is used as the conditional input for the model $D(\cdot)$ at a given timestep t, without adding any noise. Initially, the foreground image x and background image y are both randomly sampled from a Gaussian distribution. These images are then progressively refined over t iterations, where at each step, the model generates updated estimates for x and y, guided by the condition image z and the previous timestep's information. The process is iterated from $t = 1$ to $t = T$, where the generated images are progressively refined and updated based on the model's parameters, resulting in the final output. The final composited image is produced by model D as I_{out} at the last timestep.

D. More Comparison Experiments and Generative Results

D.1. Evaluation Metric

We use the Frechet Inception Distance (FID) [3] to assess the photorealism of the generated images by comparing the source image distribution with the inpainted image distributions. To measure the accuracy of correct object removal, we introduce root mean squared error (RMSE) [4], structural similarity index measurement (SSIM) [10], and

Algorithm 1 DiffDecompose Inference Algorithm

Input: Composited Image z, Trained DiffDecompose model $D(\cdot)$ at a sampling timestep t
Output: foreground image x and background image y
for epoch = 1 to t **do**
  $x_t \sim \mathcal{N}(0,\, I)$
  $y_t \sim \mathcal{N}(0,\, I)$
  $\hat{v}_x, \hat{v}_y = v_\theta(x_t, y_t, t, z, \tau)$
  $\sigma_t^2 = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\left(1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}\right)$
  $\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\cdot\hat{v}_x}{\sqrt{\bar{\alpha}_t}}\right)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\hat{v}_x+\sigma_t z_t$
  $\mathbf{y}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{y}_t-\sqrt{1-\bar{\alpha}_t}\cdot\hat{v}_y}{\sqrt{\bar{\alpha}_t}}\right)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\hat{v}_y+\sigma_t z_t$
  $I_{out} \leftarrow D(Z_T, t)$
end for

Learned Perceptual Image Patch Similarity (LPIPS) [8].

D.2. More Comparisons

We present a comprehensive comparison of our DiffDecompose and other methods on a series of tasks within our newly established benchmark AlphaBlend and publicly watermark dataset LOGO. We present two competing results in each subtask. For our proposed AlphaBlend, since the current object removal task [1, 7, 12] mainly rely on the mask-based method to achieve object removal, the task of occlusion and light removal is not reasonable for them to edit the whole image as shown in Figure 2, no matter how we tune the strengthen and change the prompt, it is still difficult to remove the rain and light. Thus, we only compare the results of the cell, glassware, and watermark, which can also be regarded as the region-level task instead of the layer-level task for these competing methods. It can be seen that when facing the transparent scenarios, the current methods directly predict the mask region with the background or prompt instead of fully utilizing the information of its transparent region. Additionally, when the occlusion region is pixel-additive, the competing method is prone to utilizing the background to generate an unreasonable object to fix the removal cell. These results suggest that when the target object is no longer a simple pixel-level coverage, introducing the region-based methods for image editing is difficult to recover accurately. In comparison, our proposed DiffDecompose can better remove the foreground layer and maintain the background layer successfully.

To further verify the effectiveness of DiffDecompose, we extend the way to the common task of watermark removal. As shown in the latest three rows, the publicly available LOGO-G, LOGO-L, and LOGO-H represent progressively challenging benchmarks for watermark removal, distinguished primarily by the transparency and size attributes of their embedded watermarks. These three public datasets can be concluded as:

LOGO-G. This dataset consists of 2,000 testing samples, characterized by gray-scale watermarks. The watermark transparency in LOGO-Gray varies broadly from 35% to 85%, encompassing both relatively faint and highly opaque watermarks. The wide transparency range introduces substantial variability, simulating scenarios where watermarks may range from subtle overlays to dominant visual artifacts. This dataset thus serves as a comprehensive testbed for evaluating models' robustness across diverse watermark visibility levels.

LOGO-L. LOGO-L restricts the watermark transparency to a narrower interval of 35% to 60%. Furthermore, the watermark size is constrained between 35% to 60% of the host image width. These constraints reflect moderate watermark prominence both in opacity and spatial extent, providing a balanced challenge for watermark removal techniques. The reduced transparency range relative to LOGO-Gray focuses the evaluation on semi-transparent watermarks, common in practical applications where watermarks are designed to be visible but non-intrusive.

LOGO-H. LOGO-H represents a more difficult subset derived from LOGO-L. It targets harder cases by increasing both watermark transparency and size, randomly selecting values from 60% to 85%. These higher transparencies and larger size ranges imply that watermarks are more visually prominent and occlusive, significantly complicating the watermark removal task. Models evaluated on LOGO-H must contend with nearly opaque and spatially extensive watermarks, testing their ability to recover underlying image content under severe occlusion. As shown in Figure 2, our method outperforms existing approaches across challenging subsets, particularly LOGO-H and LOGO-L, which involve large occlusions and foregrounds with colors closely resembling the background. Compared methods often introduce artifacts or fail to fully remove watermarks under such conditions. In contrast, our approach achieves clearer and more coherent reconstructions by jointly inferring foreground and background through a layer-wise decomposition framework. This enables effective disentanglement of semi-transparent overlays without relying on explicit masks, making our method especially robust in complex scenarios involving transparency and subtle blending.

D.3. More Generative Results

Figure 3 presents additional qualitative results for the first three subtasks of the AlphaBlend dataset: Subtask I (Translucent Flare Removal), Subtask II (Translucent Occlusion Removal), and Subtask III (Semi-transparent Watermark Removal). These extended visualizations further validate the capability of our DiffDecompose framework to disentangle complex semi-transparent and transparent layers in

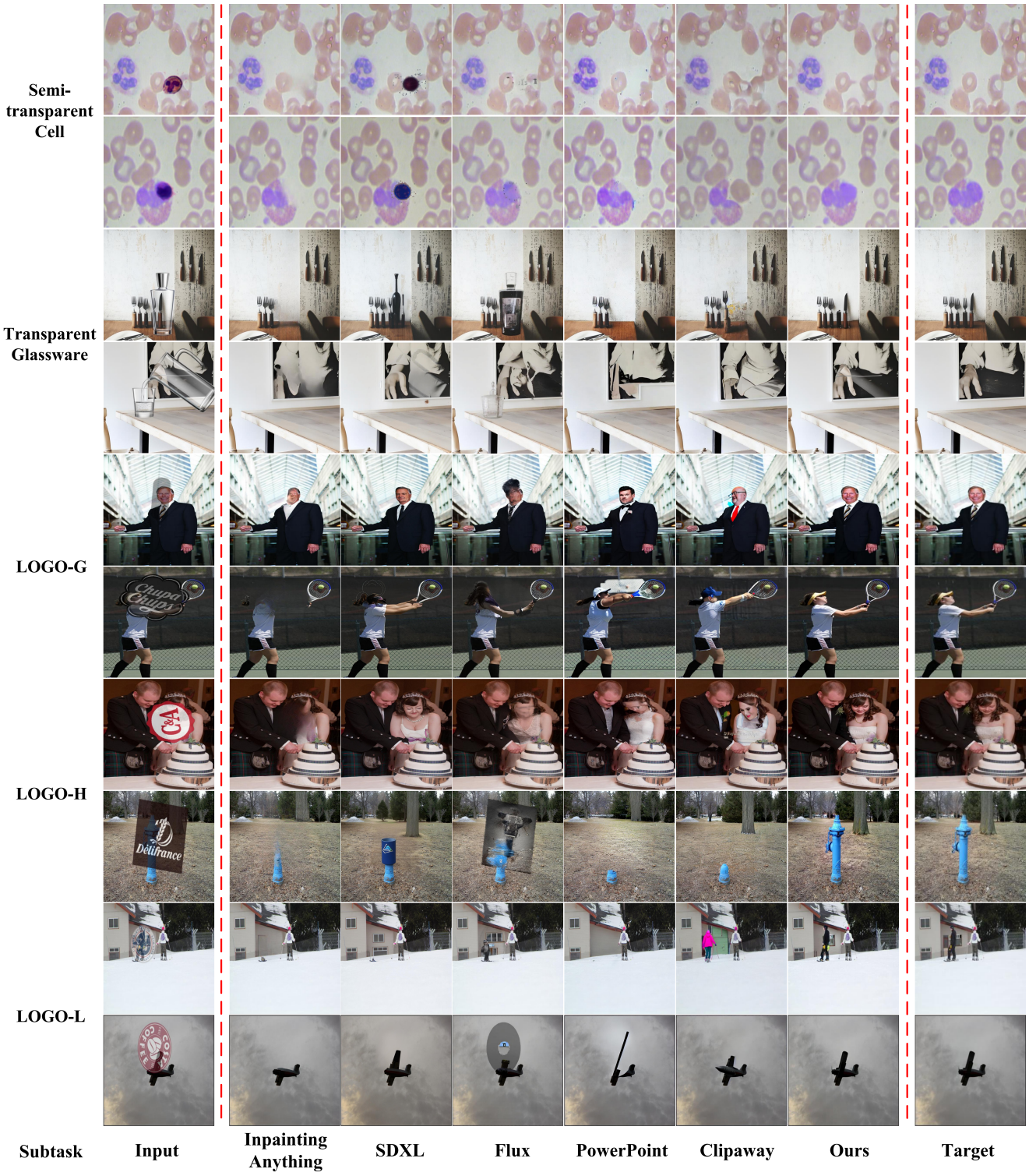

Figure 2. Qualitative results of ours and compared methods on the proposed AlphaBlend dataset and the publicly LOGO dataset. Object removal models often replace the object by predicting new pixel information instead of removing it, which fails to generate a realistic background. Our method is the only one that effectively removes objects and fills the regions in an accurate manner.

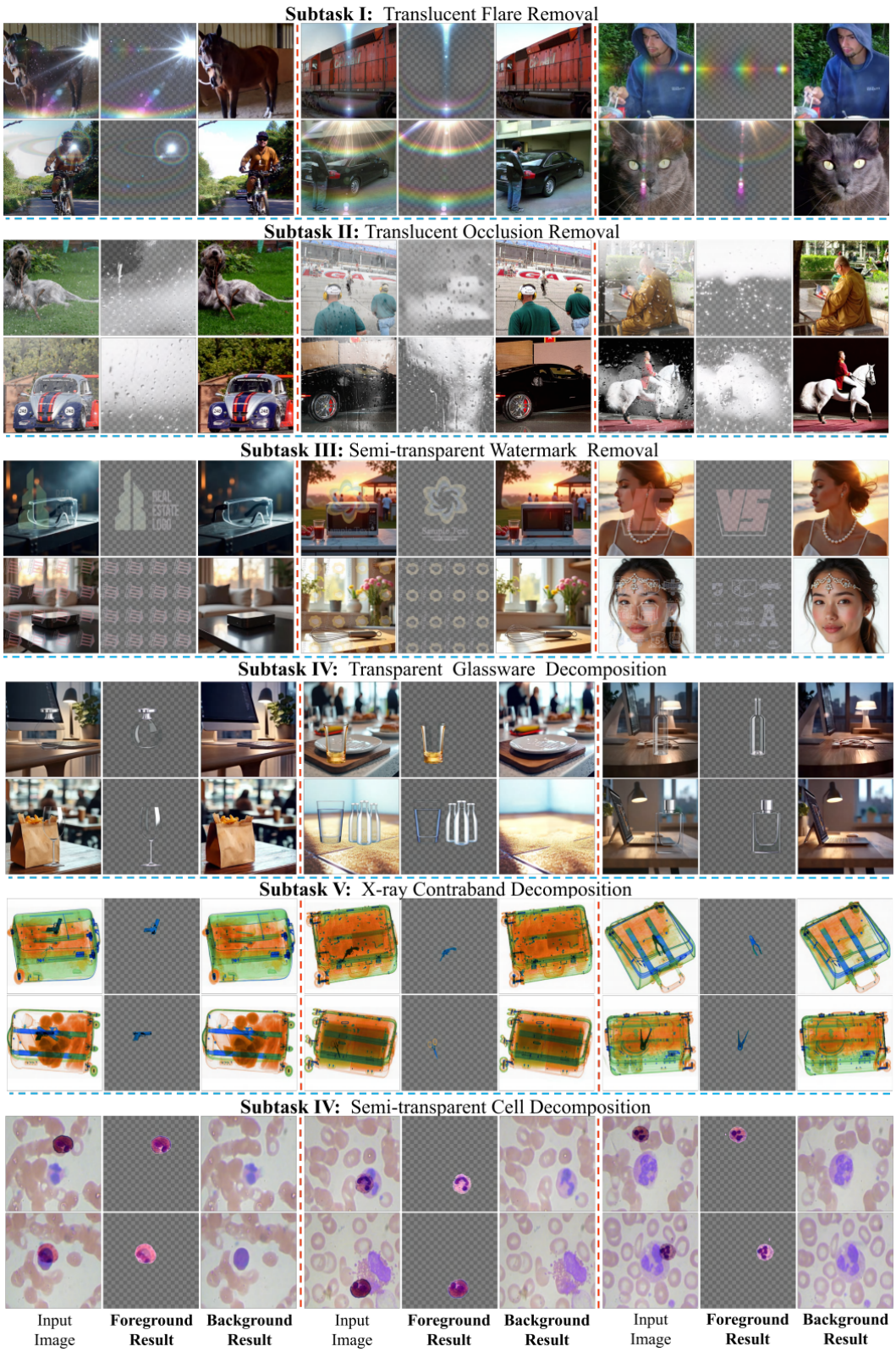

Figure 3. Additional Qualitative Results. Extended visualizations demonstrating DiffDecompose's capability in removing complex semi-transparent artifacts. The results illustrate faithful layer-wise decomposition, preserving fine details and semantic consistency across diverse scenarios.

diverse image compositions. DiffDecompose effectively removes complex lens flare while preserving scene structure and detail, separates translucent glass and reflection artifacts without disrupting background consistency, and accurately eliminates semi-transparent watermarks while maintaining underlying textures.

Furthermore, we also present more qualitative results of our proposed DiffDecompose framework on three key semi-transparent/transparent layer-wise decomposition subtasks from the AlphaBlend dataset: Subtask IV (Transparent Glassware Decomposition), Subtask V (Semi-transparent Cell Decomposition), and Subtask VI (X-ray Contraband Decomposition). These additional visualizations provide deeper insights into the robustness and generalization capabilities of our method in handling complex alpha-composited images characterized by nonlinear blending and ambiguous layer interactions. DiffDecompose isolates glass objects from varied backgrounds, accurately separates individual cells even in densely overlapping regions, and distinguishes hidden items from complex scan content while preserving contextual detail without relying on explicit masks.

As shown in Table 1 demonstrates the decomposing performance of DiffDecompose on the foreground layer. It can be seen that it successfully separates the foreground layer across all tasks, achieving low RMSE and LPIPS scores and a high SSIM score, indicating its ability to accurately reconstruct the structure and perceive the foreground. These results confirm that the method effectively preserves fine details and layer integrity even under semi-transparent and complex blending conditions. By introducing a probabilistic hierarchical decomposition framework, our model is able to recover a visually coherent and semantically consistent foreground. This highlights the advantage of modeling synthetic structures without explicit pixel-level supervision.

Table 1. Quantitative Evaluation of Foreground Separation Quality. Performance metrics, including RMSE, SSIM, LPIPS, and FID, demonstrate DiffDecompose's effectiveness in accurately extracting foreground layers across transparent glassware, X-ray contraband, and semi-transparent cell decomposition subtasks.

Subtask	**Translucent Flare Removal**				**Translucent Occlusion Removal**			
Models	RMSE↓	SSIM↑	LPIPS↓	FID↓	RMSE↓	SSIM↑	LPIPS↓	FID↓
Ours	2.6003	0.8839	0.0215	23.2082	21.3255	0.8523	0.3483	92.2930
Subtask	**Semi-transparent Watermark Removal**				**Transparent Glassware Decomposition**			
Models	RMSE↓	SSIM↑	LPIPS↓	FID↓	RMSE↓	SSIM↑	LPIPS↓	FID↓
Ours	2.3939	0.9593	0.0596	4.9729	14.9828	0.7819	0.1776	92.2640
Subtask	**X-ray Contraband Decomposition**				**Semi-transparent Cell Decomposition**			
Models	RMSE↓	SSIM↑	LPIPS↓	FID↓	RMSE↓	SSIM↑	LPIPS↓	FID↓
Ours	7.9984	0.9863	0.0276	34.7315	2.2877	0.9944	0.0082	29.270452

E. User Study

To further evaluate the perceptual quality of our model compared to the baseline methods, we conducted a user study with 30 subjects using an online voting interface, the voting interface and voting results are shown in Figures 4 and 5, respectively. For each visual scene, we presented the outputs of the six methods in a random order. Participants were asked to select the best result in terms of removal effectiveness, background integrity, and result plausibility for six representative subtasks. Our method received the highest number of votes in all six categories, demonstrating superiority over other competing baseline methods in terms of both realism and semantic coherence, without introducing unnecessary changes.

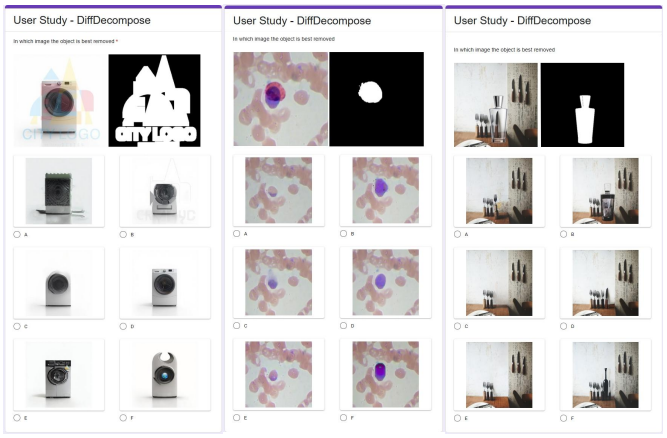

Figure 4. A user study voting interface was provided to participants. We present the performance of five competing methods on the task of watermark removal, cell separation, and glassware removal. The participants can click the alphabet to choose which method is the best and that best meets their requirements.

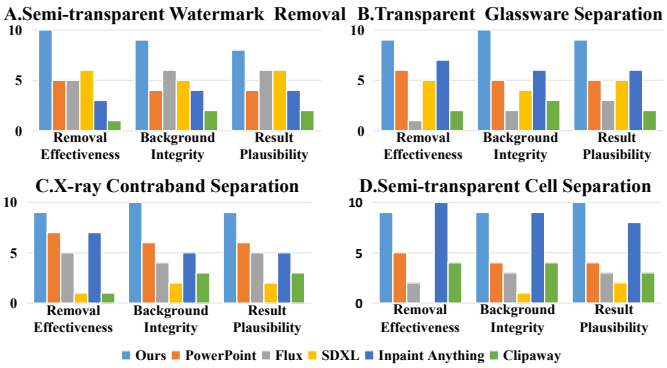

Figure 5. User study results. The voting results of DiffDecompose and the baseline method are compared on different subtasks, including removal effectiveness, background integrity, and result plausibility.

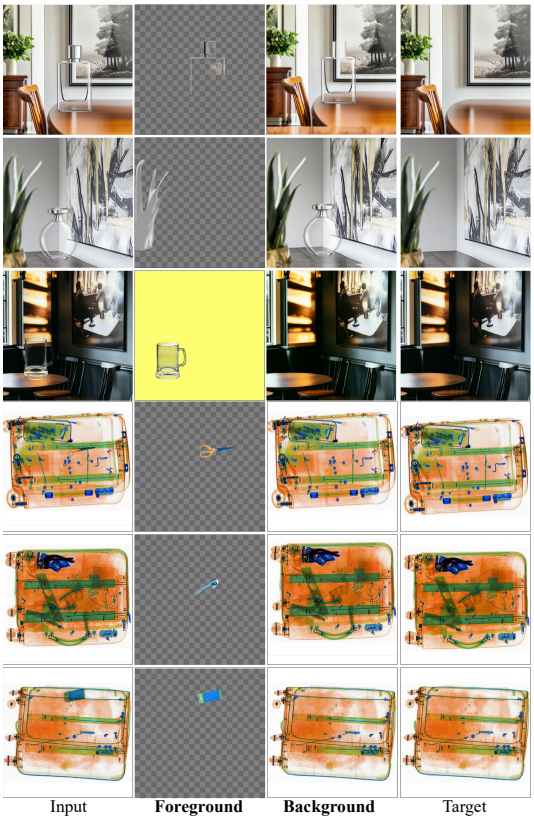

Figure 6. Results of failed decomposition by DiffDecompose.

F. Limitations and Broader Impacts

F.1. Limitations

As shown in Figure 6, although DiffDecompose performs well in various challenging semi-transparent and transparent layered decomposition tasks, it still has some limitations that deserve further study. Since DiffDecompose does not have an explicit supervision layer and fine-grained masks in the process of foreground and background separation, the pixels in the image restoration area will have pixel drift that is imperceptible to the naked eye. That is, the overall separated foreground and background information have good visual contrast, but lack pixel-level position accuracy. Therefore, we will introduce Diffusion Guidance Masks in the future to improve the network's attention to image boundaries, thereby improving the accuracy of pixel restoration.

F.2. Broader Impacts

This work may benefit applications in image editing, scientific visualization, and digital restoration by enabling accurate decomposition of semi-transparent and transparent layers. The public release of the AlphaBlend dataset and DiffDecompose code promotes reproducibility and supports broader research efforts. However, the ability to decompose image layers may also enable misuse in image manipulation or privacy invasion. We encourage responsible use and recommend the development of safeguards, particularly when extending this work to sensitive data. No personally identifiable or human subject data is involved in our experiments.

References

[1] Yiğit Ekin, Ahmet Burak Yildirim, Erdem Eren Çağlar, Aykut Erdem, Erkut Erdem, and Aysegul Dundar. Clipaway: Harmonizing focused embeddings for removing objects via diffusion models. *Advances in Neural Information Processing Systems*, 37:17572–17601, 2024. 1, 3

[2] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 1

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[4] Timothy O Hodson. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, 2022:1–10, 2022. 2

[5] Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1775–1785, 2024. 1

[6] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11450–11457, 2020. 1

[7] Suraj Patil. Sdxl inpainting. https://huggingface.co/spaces/diffusers/stable-diffusion-xl-inpainting/tree/main, 2024. 1, 3

[8] Jake Snell, Karl Ridgeway, Renjie Liao, Brett D Roads, Michael C Mozer, and Richard S Zemel. Learning to generate images with perceptual similarity metrics. In *2017 IEEE International Conference on Image Processing*, pages 4277–4281, 2017. 3

[9] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11120–11129, 2021. 1

[10] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE Signal Srocessing Letters*, 9(3):81–84, 2002. 2

[11] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2970–2979, 2017. 1

[12] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023. 1, 3