BFS: Back-to-Front Layered Image Synthesis via Knowledge Transfer

KYOUNGKOOK KANG, Samsung, Republic of Korea
GYUJIN SIM, POSTECH, Republic of Korea
SUNGHYUN CHO, POSTECH, Republic of Korea

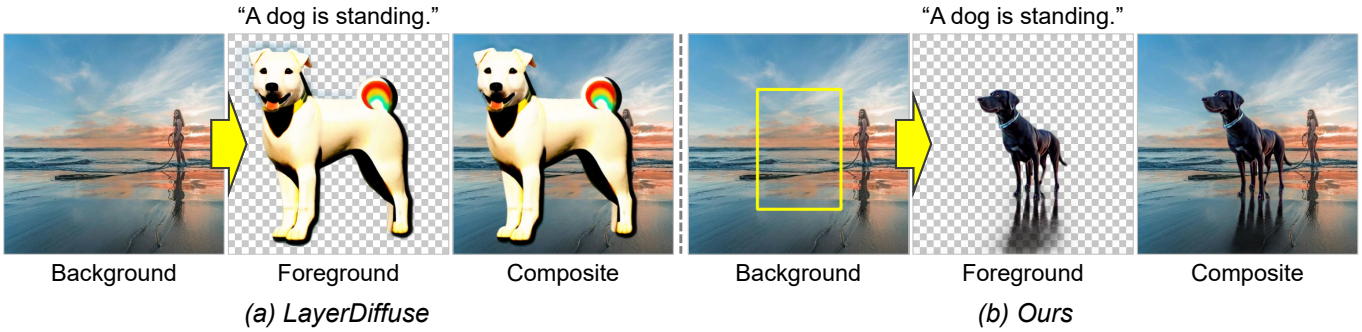

Fig. 1. Qualitative comparison of back-to-front layered image synthesis between LayerDiffuse [Zhang and Agrawala 2024] and our method. Given a background image, each method synthesizes a foreground layer and produces a composite by combining it with the background. The yellow box denotes the input mask, which is not used by LayerDiffuse. LayerDiffuse generates oversized foregrounds inconsistent with the background context, whereas our method produces contextually appropriate foreground layers with proper visual effects, including reflection and shadow.

As generative models expand the possibilities of visual content creation, layered image synthesis has emerged as a promising direction for controllable and creative editing. However, existing methods struggle to fully realize this potential. Decomposition-based methods often struggle with clean separation, while generation-based methods suffer from difficulty in training data acquisition, reducing quality and scene diversity. In this paper, we propose BFS, a novel generation-based framework for layered image synthesis. Specifically, given a background image and user guidance, BFS synthesizes a foreground layer that incorporates not only a foreground object but also its associated visual effects, such as shadows and reflections, while seamlessly harmonizing with the background to produce a coherent composite. To enable diverse and high-quality foreground layer synthesis while overcoming data scarcity, we leverage the comparatively easy-to-learn knowledge of unlayered image synthesis for the foreground synthesis. To this end, we adopt a dual-branch diffusion framework in which two interconnected branches generate a composite image and a foreground layer, respectively, enabling bidirectional knowledge transfer. Based on this framework, we propose a two-stage training scheme that utilizes a high-quality unlayered composite image dataset to effectively enhance foreground quality. Extensive experiments, including a user study, show that BFS produces high-quality layered images, consistently outperforming prior methods.

CCS Concepts: • **Computing methodologies** → **Image manipulation**.

Authors' Contact Information: Kyoungkook Kang, Samsung, Suwon, Republic of Korea, kkang831@postech.ac.kr; Gyujin Sim, POSTECH, Pohang, Republic of Korea, sgj0402@postech.ac.kr; Sunghyun Cho, POSTECH, Pohang, Republic of Korea, s.cho@postech.ac.kr.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/26/07
https://doi.org/10.1145/3799902.3811228

ACM Reference Format:
Kyoungkook Kang, Gyujin Sim, and Sunghyun Cho. 2026. BFS: Back-to-Front Layered Image Synthesis via Knowledge Transfer. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799902.3811228

1 Introduction

Layered images, typically organized as a stack of multiple RGBA layers, have long been a standard representation in professional image creation and editing. Such a representation enables independent manipulation of visual elements without affecting the rest of the image, and this non-destructive property not only streamlines iterative creative workflows but also enables dynamic reuse of visual elements across different compositions.

Recent advances in image generative models [Podell et al. 2023; Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022] have opened new opportunities for automatic layered image synthesis. According to their layer construction strategy, existing approaches can be broadly categorized into two paradigms: *decomposition-* and *generation-based* methods. *Decomposition-based* methods [Kang et al. 2025; Yang et al. 2025] break down an existing composite image into its constituent foreground and background layers, typically guided by user-provided foreground masks. However, these methods often struggle to accurately extract foreground objects together with their associated visual effects, such as shadows and reflections. Moreover, they are inherently limited to separating existing images and lack the ability to synthesize new layers to construct novel compositions.

In contrast, *generation-based* methods [Dalva et al. 2024; Huang et al. 2025a,b, 2024; Pu et al. 2025; Zhang and Agrawala 2024; Zhang et al. 2023b] directly synthesize layers from random noise and text descriptions. They produce sharp and accurate alpha masks and allow the straightforward creation of new layered images using text

description, which has driven growing research interest. However, as these approaches require large-scale layered image datasets for training, which are difficult to obtain, the development of a scalable data construction pipeline has emerged as a central challenge.

For example, Text2Layer [Zhang et al. 2023b], the MuLAn dataset [Tudosiu et al. 2024], and DreamLayer [Huang et al. 2025b] propose separation-based data construction pipelines. Starting from a composite image dataset, they extract foreground layers and then inpaint the residual regions to produce background layers. However, this process often leaves unnatural artifacts in the background, such as shadows and reflections of absent foreground objects, which in turn causes the trained model to generate incorrect outputs. Although DreamLayer [Huang et al. 2025b] filters out low-quality layered images through manual review, such reliance on human labor makes the approach unsuitable for scalable dataset construction.

Other pipelines proposed by Zhang and Agrawala [2024] and Huang et al. [2025a] start from an RGBA foreground dataset. Specifically, Zhang and Agrawala [2024] first apply outpainting outside the sampled foreground object cutouts to obtain composite images, then remove and inpaint these cutouts to produce background layers. However, the RGBA foreground dataset is limited in both diversity and scale, making the resulting composite images less representative of real-world imagery. Consequently, the generated layered images are confined to a narrow range of object categories and scene variations. Meanwhile, Huang et al. [2025a] hired professional designers to select assets from foreground and background datasets and to carefully combine them, but this labor-intensive curation is prohibitively costly. These limitations underscore the persistent challenges in layered image synthesis, especially in acquiring high-quality data at scale without substantial human involvement.

In this paper, we propose BFS (Back-to-Front Layered Image Synthesis), a novel generation-based framework that focuses on background-to-foreground (BG2FG) synthesis, i.e., synthesizing a foreground layer conditioned on a given background image. While alternative strategies exist for layered image synthesis, such as generating all layers simultaneously or synthesizing foregrounds before backgrounds, BG2FG synthesis offers two distinct advantages. First, it seamlessly extends to multi-layered image synthesis by sequentially adding new layers. Moreover, it provides a natural design workflow, where users can explore diverse layer compositions by adding objects into a scene one at a time.

To overcome the data scarcity, we leverage relatively easy-to-learn knowledge of unlayered image synthesis to guide the more challenging foreground synthesis. To this end, BFS introduces a semi-supervised learning approach based on a dual-branch diffusion architecture. Given a background image, our model employs two interconnected branches: one generates a foreground layer that harmonizes with the background, while the other produces the corresponding composite image. This architecture enables bidirectional information exchange between the branches, which facilitates knowledge transfer across modalities, improving both branches.

We train this framework in two stages. In the first stage, we use a synthetic layered image dataset, albeit of lower quality, to teach the model the structural relationships between foreground, background, and composite images, including alpha blending behavior and object-layer semantics. However, models trained solely on such data tend to produce unnatural or low-quality foreground layers. To overcome this, the second stage leverages high-quality unlayered image datasets to refine the composite generation branch. Through the shared backbone and coupled design, this refinement indirectly enhances the foreground generation branch, ultimately enabling the synthesis of high-quality layered images from limited supervision.

With extensive experiments including a user study, we demonstrate that BFS achieves high-quality layered image synthesis, where foreground and background layers are well harmonized, along with strong generalization compared to existing methods, as shown in Fig. 1. BFS also enables diverse practical applications, such as foreground layer extraction from composite and background images, reference-based foreground generation, and multi-layered image synthesis. Our main contributions can be summarized as:

- We propose BFS that enables effective BG2FG layered image synthesis while addressing the dataset challenge.
- We introduce a dual-branch diffusion model that enables cross-modal knowledge transfer, and a semi-supervised learning strategy for improved foreground synthesis.
- Extensive experiments demonstrate that BFS outperforms existing methods in both visual quality and generalization, showing strong applicability to real-world scenarios.

2 Related Work

Layered Image Synthesis. Motivated by the practical value of layered image representations, interest in their automatic generation has grown rapidly. In this section, we review the architectural designs adopted by prior methods. Beyond the early GAN-based [Bae et al. 2022] and CLIP-based [Bar-Tal et al. 2022] attempts, most recent methods leverage the generative power of pretrained diffusion models via fine-tuning. Decomposition-based methods [Kang et al. 2025; Yang et al. 2025] adapt the input/output layers of diffusion models to decompose a composite image into foreground and background layers.

Generation-based methods directly synthesize layers from random noise and text descriptions. Text2Layer [Zhang et al. 2023b] introduces a unified latent space in which foreground and background are jointly embedded, enabling both layers to be generated in a single pass. Subsequent methods [Dalva et al. 2024; Huang et al. 2025a,b, 2024; Zhang and Agrawala 2024] have established a dominant paradigm of multi-pass joint synthesis, allocating a dedicated generative pathway to each layer. For instance, LayerDiff [Huang et al. 2024] synthesizes layers from layer-specific text prompts while promoting both intra- and inter-layer interactions via a collaborative attention block, and LayerDiffuse [Zhang and Agrawala 2024] leverages layer-specific LoRA adapters for each layer and aggregates all attention vectors across all pathways to form a unified model.

More recent works [Dalva et al. 2024; Huang et al. 2025a,b] augment the design with an additional composite branch, leveraging cross-attention maps to inject global scene context into the individual layers. Nevertheless, they remain tied to U-Net–based diffusion backbones and depend heavily on cross-attention maps. In contrast, the only DiT-based work, ART [Pu et al. 2025], adopts a fundamentally different strategy. It first generates a global layout, assigns

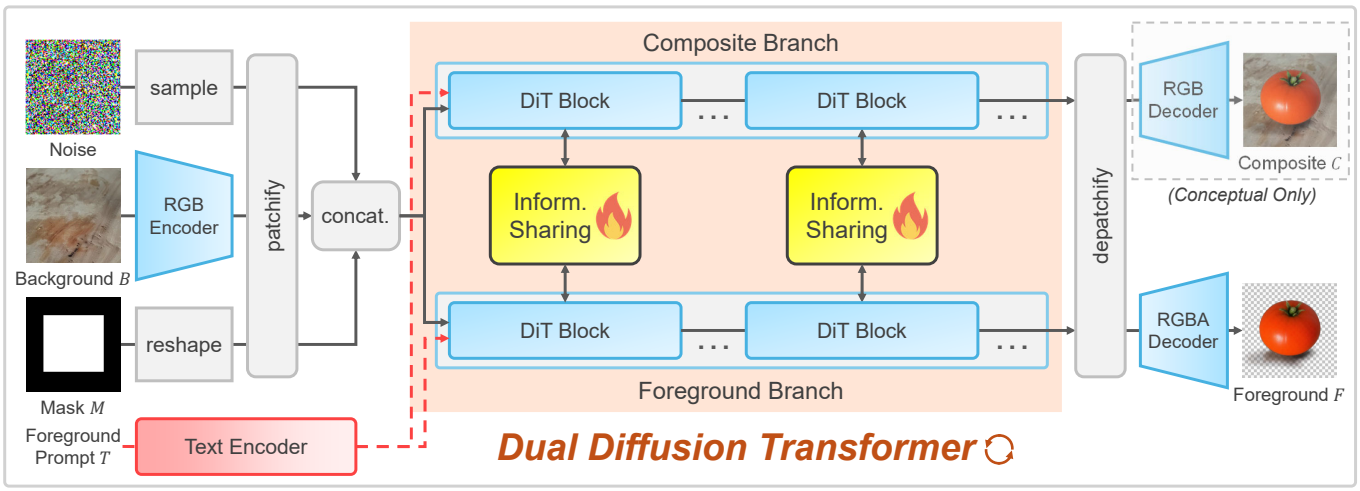

Fig. 2. Overall framework of BFS. Given a background image B, a bounding box M, and a foreground prompt T, BFS generates a foreground RGBA layer F through a bidirectionally interconnected dual-branch diffusion transformer. The composite branch produces a composite latent for knowledge transfer, but its output is not decoded. Only the foreground latent is decoded to obtain the final result.

latent tokens to spatial regions, and synthesizes all layers simultaneously with a diffusion transformer. While this design enables efficient region-wise generation, it inherently limits the ability to capture holistic visual effects spanning across multiple regions.

Object Insertion. Another closely related line of research is object insertion, which integrate an object into a target scene while ensuring object-level plausibility (pose, scale, and identity) and scene-level consistency (lighting, shadows, reflections, and other visual effects). Recently, large-scale image generative models have motivated *single framework solutions* that address all the subproblems simultaneously, directly producing composite images in a single synthesis. Among them, training-free methods [Lu et al. 2023b; Tewel et al. 2024] demonstrate that high-quality insertion results can be achieved by guiding the diffusion generation process. A more dominant direction is to fine-tune diffusion models to insert a reference image into a given background [Chen et al. 2024; Song et al. 2023, 2024; Yang et al. 2023; Zhang et al. 2023a]. To improve realism, subsequent works curate interaction-aware data or exploit object-removal pairs [Canberk et al. 2024; Canet Tarrés et al. 2024; He et al. 2024; Huang et al. 2025c; Kim et al. 2025; Song et al. 2025; Winter et al. 2024a; Yu et al. 2025]. Recent extensions also explore multi-object insertion and multi-view identity cues [Lu et al. 2023a; Tarrés et al. 2025; Winter et al. 2024b], and layout-first then inpaint formulations [Singh et al. 2024; Yun et al. 2024].

However, most object insertion methods concentrate on composite image generation rather than learning disentangled representations. As a result, the inserted objects cannot be easily separated, reused, or independently edited after synthesis. In contrast, our approach explicitly generates a foreground RGBA layer which is an editable and reusable representation of the inserted object.

3 Dual-Branch Diffusion Architecture

As introduced in Sec. 1, we focus on the BG2FG problem in layered image synthesis by proposing BFS, which is illustrated in Fig. 2. Given a background image B, a bounding-box mask M, and a foreground text description T, it generates a foreground RGBA layer $F = (F_{rgb}, F_\alpha)$ that contains the object specified by T within M. The composition follows the standard alpha blending equation:

$$C = F_\alpha \cdot F_{rgb} + (1 - F_\alpha) \cdot B, \tag{1}$$

where F_{rgb} and F_α are the RGB color channels and corresponding opacity map of F, respectively, and $\cdot$ is element-wise multiplication.

To enable post-editing of the synthesized layered image, we define the foreground representation to satisfy the following properties: (1) it includes not only the foreground object but also its associated visual effects, such as shadows and reflections, and (2) it naturally harmonizes with the background when composited. These properties allow lightweight and immediate editing in typical real-world scenes, where such effects remain largely consistent under moderate foreground transformations (e.g., repositioning or replacement).

To synthesize such foreground layers, BFS is to leverage the relatively easy-to-learn knowledge of composite image synthesis to guide the more challenging foreground synthesis, thereby alleviating the need for complex training data construction. To achieve this, we adopt a dual-branch diffusion architecture that jointly generates a composite image and its foreground layer, with bidirectional information exchange ensuring semantic alignment between branches.

Specifically, we instantiate two parallel copies of a diffusion transformer and introduce an *information-sharing* module based on a symmetric cross-attention layer. To detail the information-sharing module, let H^C and H^F be the intermediate features before the self-attention layer at each transformer block of the composite and foreground pathways, respectively. We compute two cross-attentions by querying one pathway with keys and values from the other:

$$Z^{C\to F} = \text{softmax}\left(\frac{Q^F(K^C)^\top}{\sqrt{d}}\right)V^C, \quad Z^{F\to C} = \text{softmax}\left(\frac{Q^C(K^F)^\top}{\sqrt{d}}\right)V^F, \tag{2}$$

where Q, K, and V denote the query, key, and value embeddings of each branch, while superscripts F and C indicate the foreground and composite pathways, respectively. d is the embedding dimension.

For computing the cross-attention embeddings, we use shared and frozen base matrices W_Q, W_K, and W_V, initialized from the backbone's self-attention weights, and introduce two *direction-specific* LoRAs $\{\Delta W^{C\to F}_{\{\cdot\}}, \Delta W^{F\to C}_{\{\cdot\}}\}$ to modulate these matrices. The resulting feature embeddings are computed as:

$$\begin{aligned} Q^C &= H^C\left(W_Q + \Delta W_Q^{F\to C}\right),\ Q^F = H^F\left(W_Q + \Delta W_Q^{C\to F}\right), \\ K^C &= H^C\left(W_K + \Delta W_K^{C\to F}\right),\ K^F = H^F\left(W_K + \Delta W_K^{F\to C}\right), \\ V^C &= H^C\left(W_V + \Delta W_V^{C\to F}\right),\ V^F = H^F\left(W_V + \Delta W_V^{F\to C}\right). \end{aligned} \tag{3}$$

The two cross-attention messages are concatenated and passed through a lightweight MLP $g(\cdot)$, whose outputs are split into two residuals ΔH^C and ΔH^F for each branch:

$$[\Delta H^C, \Delta H^F] = g\big(\text{concat}(Z^{F\to C}, Z^{C\to F})\big). \tag{4}$$

We then integrate these residuals into each branch by adding them to the output of the self-attention layer of each branch:

$$\widetilde{H}^C = \text{SelfAttn}(H^C) + \Delta H^C,\ \widetilde{H}^F = \text{SelfAttn}(H^F) + \Delta H^F. \tag{5}$$

In our implementation, we adopt Flux-Fill [Labs 2024] as our backbone transformer, which is an image-inpainting diffusion transformer selected for its task affinity. Keeping the backbone frozen, we train only the two *direction-specific information-sharing* LoRAs ($\{\Delta W_\cdot^{C\to F},\ \Delta W_\cdot^{F\to C}\}$) and a lightweight MLP $g(\cdot)$. To encode background and composite inputs into the latent space, we use the pretrained RGB VAE of Flux-Fill, while the RGBA foreground layer is encoded and decoded using an RGBA VAE fine-tuned on our dataset. Additional details on RGBA VAE fine-tuning and other implementation details including LoRA configurations are provided in the supplementary material.

At inference time, the diffusion denoising process operates in the latent space of the pretrained VAE. A random noise tensor is first channel-wise concatenated with the conditioning inputs (the background latent and a bounding box mask reshaped following Flux-Fill). The concatenated tensor is then duplicated along the batch dimension and passed through the diffusion transformer with text embeddings computed from foreground descriptions. After iterative denoising, the model produces a composite latent and a foreground latent. Finally, only the foreground latent is decoded into the image domain using the RGBA VAE.

4 Two-stage Semi-Supervised Learning

To train BFS, the most straightforward dataset would comprise foreground, background, composite images, and bounding-box masks. Yet, as discussed in Sec. 1, constructing a layered image dataset in practice is highly challenging. To overcome this challenge, we propose a two-stage semi-supervised learning strategy that avoids explicit reliance on ground-truth foreground layers. In brief, we first train BFS on a simulation-based synthetic dataset that provides all modalities, allowing the model to learn their structural relationships. We then fine-tune it on a high-quality unlayered image dataset to enhance the composite synthesis, which in turn improves the foreground synthesis through the information-sharing module. In the following sections, we provide the details of the datasets and the training losses used in each stage.

4.1 Representation Learning with Compositional Consistency

BFS first learns modality-specific representations and their compositional relationships using a synthetic dataset composed of foreground, background, and composite images, along with bounding-box masks. To construct the dataset, we adopt an approach similar to LayerDecomp [Yang et al. 2025] on the RORem object removal dataset [Li et al. 2025], which provides composite images, object masks, and background images where both objects and their visual effects have been removed.

For each composite image C, we first extract a foreground layer F from the object mask M using a matting technique [Yao et al. 2024], then augment it to include simulated shadows using an internal shadow generation network, which takes the foreground layer placed on a white background as input and renders shadow effects into the image. The augmented foreground layer $\hat{F}$ is recomposited with the background layer B to form a new composite image $\hat{C}$. Finally, we replace the original object mask with a bounding-box mask $\hat{M}$, resulting in a training corpus of $\{\hat{F}, B, \hat{M}, \hat{C}\}$, as shown in Fig. 3 (a). We provide the details of the shadow generation network and visual examples of the dataset construction pipeline in the supplementary material. For notational simplicity, we drop the hat hereafter.

Our data construction pipeline produces image layers with precise pixel-level alignment and embeds simulated visual effect into the foreground layer, even though these effects are less realistic and not harmonized with the background layer. Unlike LayerDecomp [Yang et al. 2025], which constructs composites by randomly pairing foregrounds with backgrounds, our method derives foregrounds directly from real composite images, thereby offering more reliable supervision for object placement and scale.

Using the synthetic dataset, we employ a flow-matching loss. Let z_F, z_B, and z_C denote the latent representations of the foreground, background, and composite images, respectively. At each training iteration, we sample Gaussian noise $z_0 \sim \mathcal{N}(0, I)$ and linearly interpolate it with each data latent $z_1 \in \{z_F, z_C\}$ at a random time step $t \sim \mathcal{U}(0,1)$: $z^{(t)} = (1-t)z_0 + tz_1$ The flow-matching loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{flow}} = \mathbb{E}_{t,z_0,z_1}\Big[&\left\|v_C^* - v_\theta^C(z_C^{(t)}, z_F^{(t)}, t, \kappa)\right\|_2^2 \\ &+\left\|v_F^* - v_\theta^F(z_C^{(t)}, z_F^{(t)}, t, \kappa)\right\|_2^2\Big], \end{aligned} \tag{6}$$

where v_C^* and v_F^* are the ground-truth velocity fields (given by $v^* = z_1 - z_0$). v_θ^C and v_θ^F are velocity fields predicted by the composite and foreground branches, respectively. Note that v_θ^C and v_θ^F are computed from both $z_C^{(t)}$ and $z_F^{(t)}$, as the composite and foreground branches are connected through information-sharing modules. Here,

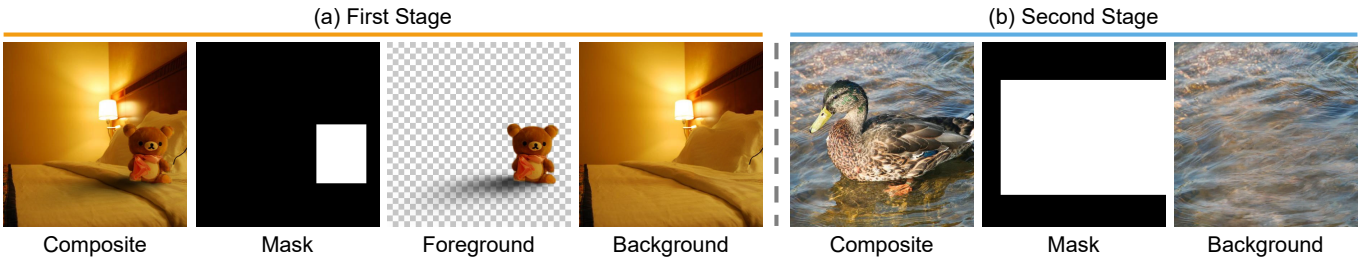

Fig. 3. Sample images from the training corpora used in each stage.

κ represents the conditioning inputs, including the foreground text description T, the background latent z_B, and a reshaped mask M.

To further enforce compositional consistency between the branches, we introduce a composition loss, which is defined as:

$$\mathcal{L}_{\text{comp}} = \left\|\Phi(\hat{z}_F, z_B) - \hat{z}_C\right\|_1, \tag{7}$$

where $\hat{z}_C$ and $\hat{z}_F$ are the model's one-step denoised estimates at time t, (i.e., the predicted clean latents from $z_C^{(t)}$ and $z_F^{(t)}$). The function Φ composites the foreground latent $\hat{z}_F$ and the background latent z_B in the latent space. We implement Φ as a neural network, pre-trained separately and then fixed during the training of our framework. More details of Φ is provided in the supplementary material. Finally, the overall objective of the first stage is defined as:

$$\mathcal{L}_{\text{first}} = \mathcal{L}_{\text{flow}} + 0.1\mathcal{L}_{\text{comp}}. \tag{8}$$

4.2 Realism Enhancement with Distribution Regularization

In the second stage, we enhance the realism of the generated images using a dataset of natural, unlayered images that capture diverse real-world effects. Specifically, we construct high-quality paired background and composite images and use them to supervise only the composite branch. The resulting improvements then propagate to the foreground branch through the information-sharing modules, aligning the foreground with the refined composite and calibrating its visual effects.

To be specific for the dataset construction, we remove foreground objects from composite images of the RORem dataset [Li et al. 2025] to generate corresponding background images as shown in Fig. 3 (b). Although the RORem dataset already provides background images obtained by inpainting the masked regions, these often still contain residual visual effects of the foreground objects. To obtain cleaner backgrounds, we instead employ ObjectClear [Zhao et al. 2025], a recent object-removal method that removes both objects and their associated effects. This construction pipeline is scalable, which makes it easy to further expand the training corpus.

Based on the dataset, we define a realism-boosting loss that supervises only the composite branch:

$$\mathcal{L}_{\text{real}} = \mathbb{E}_{t,z_0,z_1}\left[\left\|v_C^* - v_\theta^C(z_C^{(t)}, z_S^{(t)}, t, \kappa)\right\|_2^2\right], \tag{9}$$

where $v^* = z_1 - z_0$. As no ground-truth foreground layer F is available, we instead use a *surrogate* input S to compute the surrogate latent $z_S^{(t)}$ for the input of the foreground branch, where S is obtained by applying an image matting network [Yao et al. 2024] to the composite image C based on the mask M. We empirically find that this surrogate input has negligible influence on learning dynamics.

While the composite supervision improves realism, relying on it alone may cause the foreground distribution to drift away from the distribution established in the previous stage, degrading the quality of the foreground layer. To mitigate this, we introduce additional supervision for the foreground branch using the synthetic dataset of the previous stage. Although the dataset is not fully realistic, it benefits from the pixel-wise alignment across $\{F, B, C\}$. Using this dataset, we introduce a foreground regularization loss defined with only the foreground branch term of the flow-matching objective in Eq. (6):

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{t,z_0,z_1}\left[\left\|v_F^* - v_\theta^F(z_C^{(t)}, z_F^{(t)}, t, \kappa)\right\|_2^2\right]. \tag{10}$$

During training iterations, we alternate between optimizing the realism-boosting loss and the foreground regularization loss, each combined with the composition loss. Specifically, we optimize $\mathcal{L}_{\text{real}} + 0.5\mathcal{L}_{\text{comp}}$ with an 80% probability and $\mathcal{L}_{\text{reg}} + 0.5\mathcal{L}_{\text{comp}}$ with a 20% probability at each iteration.

5 Experiments

5.1 Layered Image Synthesis Evaluation

Baselines. We compare the quality of the layered images generated by BFS and existing approaches on multiple datasets, including SAM-FB [He et al. 2024], RORem [Li et al. 2025], and AnyInsertion [Song et al. 2025]. Each dataset provides background images and masks specifying the regions where foreground objects should be placed. For consistency, all masks are converted into bounding-box form. All input images (backgrounds and masks) are resized such that the shorter side is 512 pixels before being fed into the models. For the target foreground text descriptions, we use the captions provided by the SAM-FB and AnyInsertion datasets, while those for the RORem dataset are generated using BLIP2 [Li et al. 2023].

Since decomposition-based methods are not designed for BG2FG synthesis, we instead evaluate their outputs by applying the extracted layers to the ground-truth composite images. Specifically, we first compute foreground masks using SAM2 [Ravi et al. 2024] from the bounding-box masks, and estimate foreground layers using LayeringDiff [Kang et al. 2025]. We then estimate the background layers using ObjectClear [Zhao et al. 2025], since LayeringDiff often leaves residual visual effects in the background. We refer to this combination as LD+OC. For LayeringDiff and ObjectClear, we

Table 1. Quantitative evaluation of existing layered image synthesis methods on RORem dataset [Li et al. 2025]. †Note that KID may yield small negative values due to sampling variance. These should be interpreted as values close to zero.

	FG	Composite (or Background for LD+OC)				
Method	CLIP↑	MUSIQ↑	MANIQA↑	$KID_{\times 1000}$↓	DINO↑	HPSv3↑
LD+OC	0.701	63.485	0.382	**-0.2**†	**0.961**	2.91
LayerDiffuse	**0.770**	67.221	0.425	11.0	0.467	-3.03
Ours	0.721	**67.886**	**0.437**	0.2	0.875	**3.70**

Table 2. Results of the user study on 20 examples, averaged over 18 participants. Higher scores indicate better quality (1–5).

Metric	LD+OC	LayerDiffuse	BFS
(1) FG–Text Alignment	**4.02**	2.96	3.99
(2) FG–Visual Effects	3.49	2.33	**3.90**
(3) FG–Image Quality	3.61	2.29	**3.88**
(4) Composite–Placement	4.11	1.82	**4.32**
(5) Composite–Harmonization	3.04	1.28	**3.68**
(6) Composite–Image Quality	3.33	1.64	**3.74**
Average	3.60	2.05	**3.92**

use the code provided by the authors. We do not include LayerDecomp [Yang et al. 2025] in our comparison as its source code is not publicly available. However, the quality of its background estimates are expected to be comparable to ObjectClear, given the close similarity in their training strategies.

For generation-based methods, we only compare with LayerDiffuse [Zhang and Agrawala 2024], as it is the only method with publicly available code. Specifically, we adopt its best-performing SDXL-based two-stage variant, which first generates a composite image and then estimates the foreground layer conditioned on the background and the composite. For other generation-based models, we provide a discussion based on their reported results in the supplementary material.

Qualitative Comparison. Fig. 6 presents a qualitative comparison. LD+OC fails to capture fine foreground details (e.g., under the train body), and the recovered background does not faithfully restore the rail structures. The foreground objects generated by LayerDiffuse are oversized and inconsistent with the background context. This indicates that the synthetic dataset used for its training does not generalize well to real-world backgrounds, leaving the model uncertain about how to position objects, and simply producing large, centrally placed ones that follow the typical distribution of RGBA images. In contrast, our method generates foreground objects that are contextually appropriate, visually consistent, and equipped with proper visual effects, yielding more coherent composites.

Quantitative Comparison. Quantitative evaluation of generated layered images is inherently difficult, as ground-truth layered images are not available. In this work, we evaluate both the generated foreground layer and the composite image obtained by combining the generated foreground with the input background. For decomposition-based methods, whose input is a composite image, we evaluate their estimated background layers instead. For foreground evaluation, we use CLIP score [Hessel et al. 2021] to measure alignment between the generated (or decomposed) foreground and the input foreground caption. For composite (or background) evaluation, we employ non-reference aesthetic quality metrics (MUSIQ [Yu et al. 2025] and MANIQA [Yu et al. 2025]), as well as KID [Bińkowski et al. 2018] to measure distributional similarity against the target distribution. We further report the DINO score [Oquab et al. 2023] to directly compare with the ground-truth. Finally, to quantify human preference, we adopt the HPSv3 [Ma et al. [n. d.]], which is designed to align with human judgments. Since HPSv3 requires a text description as input, we use captions generated by BLIP2 [Li et al. 2023] on ground-truth.

We report results on 2,000 images from the RORem dataset in Tab. 1, while results on other datasets are provided in the supplementary material. LayerDiffuse [Zhang and Agrawala 2024] attains the highest CLIP scores, as it tends to generate large foreground objects that dominate the image and are thus favored by the metric. However, its composite evaluation scores are substantially worse than other methods, especially in KID and DINO, indicating that the generated composites deviate significantly from the ground-truth distribution. LD+OC [Kang et al. 2025; Zhao et al. 2025] achieves strong performance in both KID and DINO, highlighting its strength in background estimation. Our method achieves a higher CLIP score than LD+OC, while also outperforming LayerDiffuse by a large margin in composite evaluation. Moreover, our method achieves the highest HPSv3 score among all methods.

User Study. We designed a human preference study recruiting 18 volunteers from our institution and 20 test cases. In each test case, participants were presented with the foreground prompt along with the generated foreground, background, and composite images produced by LD+OC [Kang et al. 2025; Zhao et al. 2025], LayerDiffuse [Zhang and Agrawala 2024], and BFS. Participants rated six aspects: (1) alignment between the foreground and text description, (2) correctness of visual effects (e.g., shadows and reflections) in the foreground, (3) visual quality of the foreground, (4) appropriateness of object placement in the composite, (5) harmonization between the foreground and background, and (6) overall visual quality of the composite. As summarized in Tab. 2, BFS achieves consistently high ratings across all criteria, demonstrating a clear preference among participants. The detailed questionnaire is provided in the supplementary material.

5.2 Comparison with Object Insertion Methods

BFS is conceptually related to naturally adding new objects into a given scene. To evaluate this capability, we compare it against two recent models designed for similar purposes: the image inpainting model Flux-Fill [Labs 2024] and the object insertion method PaintByInpaint [Wasserman et al. 2025]. For Flux-Fill, we adopt the default setting. Since PaintByInpaint does not take an input mask and uses an instruction-based text description, we prepend the word "add" to the foreground caption before providing it to the model.

Fig. 7 presents a qualitative comparison of the composites generated by these baselines and our approach. For reference, we also

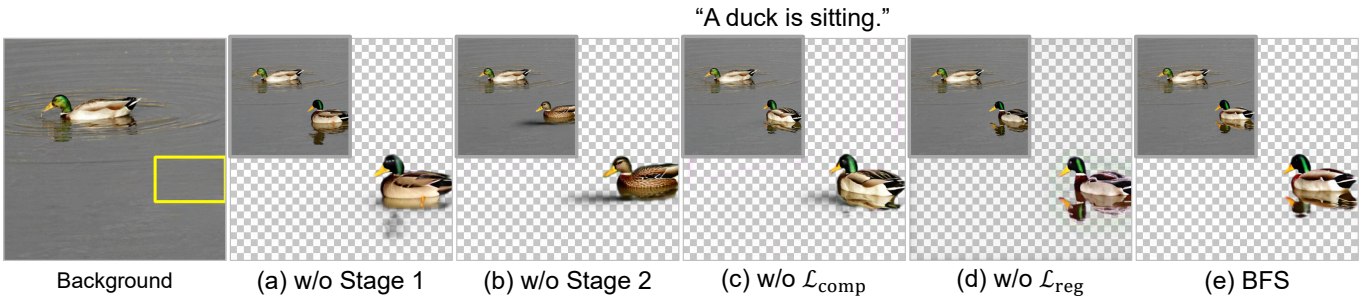

Fig. 4. Ablation study. Qualitative comparison of BFS with different components removed. The yellow box denotes the input mask and the inset in the top-left shows the output of the composite branch for reference.

Table 3. Quantitative comparison with an object insertion method PaintByInpaint [Wasserman et al. 2025] and an image inpainting method Flux-Fill [Labs 2024] on RORem datset [Li et al. 2025].

Method	MUSIQ↑	MANIQA↑	KID$_{\times 1000}$↓	DINO↑	HPSv3↑
PaintByInpaint	65.516	0.358	0.889	0.824	1.65
Flux-Fill	66.132	0.425	0.934	0.857	2.74
Ours	**67.886**	**0.437**	**0.2**	**0.875**	**3.70**

show the foreground layers produced by BFS. PaintByInpaint often struggles to place objects at semantically suitable locations, while FluxFill generates realistic images but performs editing strictly within the input bounding box, often resulting in composites that lack proper visual effects. In contrast, our method synthesizes high-quality composite images, while additionally producing explicit and reusable foreground layers.

Tab. 3 summarizes the quantitative evaluation of the generated composite images on the RORem dataset [Li et al. 2025]. Our method achieves the best scores across all metrics, demonstrating that BFS produces natural and coherent composites that are on par with state-of-the-art insertion models trained solely on real-world supervision.

5.3 Ablation Study

We conduct an ablation study to assess the contribution of each component in BFS. Fig. 4 presents qualitative comparisons of different model variants given the same background and mask (highlighted in yellow), with the outputs of the composite branch in the inset. Without Stage 1 representation learning, directly training Stage 2 degrades foreground quality, yielding inferior reflection quality. Omitting Stage 2 generates composites with unnatural shadows that propagate to the foreground layer. Removing the composition loss breaks consistency between the composite and foreground, causing unnatural reflections. Excluding the regularization step in Stage 2 collapses the RGBA distribution, yielding green halo regions around the foreground. In contrast, our full model successfully aligns both branches and produces high-quality RGBA layers.

5.4 Further Applications

BFS can be extended to other applications as shown in Fig. 5. First, it enables *foreground extraction* from existing images by feeding a noise-perturbed composite into the composite branch. Unlike conventional matting, this produces a foreground layer that preserves both the object and its visual effects (a). Second, it supports *reference-based foreground synthesis* by replacing the foreground

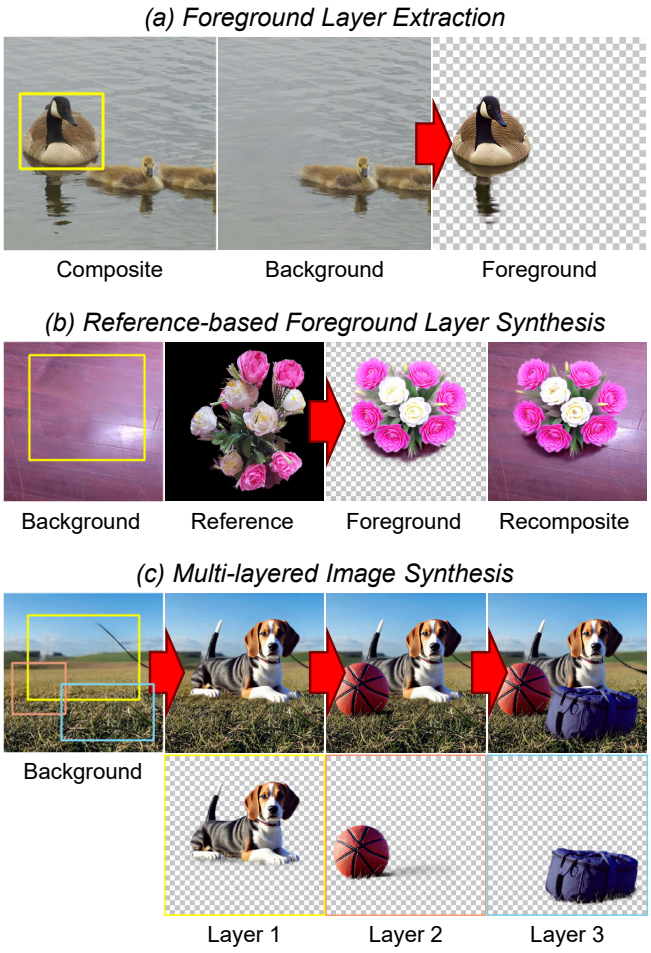

Fig. 5. (a) Foreground layer extraction from a composite image given its corresponding background image with the target foreground removed. (b) Reference-based foreground layer synthesis guided by a reference image. (c) Multi-layered image synthesis achieved through the sequential application.

text embedding with that of a reference image via a Flux-Redux adapter, allowing the synthesized foreground to inherit the reference's characteristics (b). Lastly, BFS extends to *multi-layered image synthesis*, generating scenes with multiple foreground layers (c). More results appear in the supplementary material.

5.5 Post-editing of Synthesized Layered Images

The layered images synthesized by BFS naturally support flexible post-hoc editing. Fig. 8 illustrates several examples where input background images (a) are first extended into a richer scene by adding multiple layers with BFS (b), and then further manipulated by editing each individual layer (c). All edits are performed with standard tools in Adobe Photoshop, demonstrating that the synthesized layered representations integrate seamlessly into conventional raster-graphics workflows and enable practical image generation.

6 Conclusion

We present BFS, a novel generation-based framework for high-quality layered image synthesis. To overcome data scarcity, we transfer the relatively easy-to-learn knowledge of composite synthesis to foreground synthesis through a dual-branch framework that enables bidirectional knowledge exchange between the two branches. We also propose a two-stage training scheme that utilizes a high-quality unlayered image dataset to further enhance foreground quality. Extensive experiments demonstrate that BFS produce high-quality layered images, outperforming prior methods.

Limitations. BFS has certain limitations. Its dual-branch design increases inference time, as generating a single output with BFS takes 25 seconds compared to 12 seconds with Flux-Fill backbone. In addition, our training data synthesis pipeline relies on an object-centric matting formulation over general images, which introduces two limitations. First, the relative scarcity of high-quality alpha mattes for transparent objects makes transparent object insertion challenging. We provide further discussion and examples in the supplementary material. Second, this formulation struggles to represent spatially diffuse or non-object-centric phenomena, making scene-wide effects such as fog or raindrops difficult to generate. Finally, although BFS can synthesize plausible visual effects, these effects are not guaranteed to be physically accurate. In future work, we plan to incorporate rendering toolchains that can model more complex and physically grounded effects.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH); No. RS-2024-00395401, Development of VFX Creation and Combination Using Generative AI; No. RS-2024-00457882, AI Research Hub Project).

References

Jeongmin Bae, Mingi Kwon, and Youngjung Uh. 2022. Furrygan: High quality foreground-aware image synthesis. In *European Conference on Computer Vision*.

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. 2022. Text2live: Text-driven layered image and video editing. In *European conference on computer vision*.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401* (2018).

Alper Canberk, Maksym Bondarenko, Ege Ozguroglu, Ruoshi Liu, and Carl Vondrick. 2024. Erasedraw: Learning to insert objects by erasing them from images. In *European Conference on Computer Vision*.

Gemma Canet Tarrés, Zhe Lin, Zhifei Zhang, Jianming Zhang, Yizhi Song, Dan Ruta, Andrew Gilbert, John Collomosse, and Soo Ye Kim. 2024. Thinking outside the bbox: Unconstrained generative object compositing. In *European Conference on Computer Vision*.

Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. 2024. Anydoor: Zero-shot object-level image customization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Yusuf Dalva, Yijun Li, Qing Liu, Nanxuan Zhao, Jianming Zhang, Zhe Lin, and Pinar Yanardag. 2024. Layerfusion: Harmonized multi-layer text-to-image generation with generative priors. *arXiv preprint arXiv:2412.04460* (2024).

Jixuan He, Wanhua Li, Ye Liu, Junsik Kim, Donglai Wei, and Hanspeter Pfister. 2024. Affordance-aware object insertion via mask-aware dual diffusion. *arXiv preprint arXiv:2412.14462* (2024).

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718* (2021).

Dingbang Huang, Wenbo Li, Yifei Zhao, Xinyu Pan, Yanhong Zeng, and Bo Dai. 2025a. PSDiffusion: Harmonized Multi-Layer Image Generation via Layout and Appearance Alignment. *arXiv preprint arXiv:2505.11468* (2025).

Junjia Huang, Pengxiang Yan, Jinhang Cai, Jiyang Liu, Zhao Wang, Yitong Wang, Xinglong Wu, and Guanbin Li. 2025b. DreamLayer: Simultaneous Multi-Layer Generation via Diffusion Mode. *arXiv preprint arXiv:2503.12838* (2025).

Junjia Huang, Pengxiang Yan, Jiyang Liu, Jie Wu, Zhao Wang, Yitong Wang, Liang Lin, and Guanbin Li. 2025c. DreamFuse: Adaptive Image Fusion with Diffusion Transformer. *arXiv preprint arXiv:2504.08291* (2025).

Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. 2024. LayerDiff: Exploring Text-guided Multi-layered Composable Image Synthesis via Layer-Collaborative Diffusion Model. *arXiv preprint arXiv:2403.11929* (2024).

Kyoungkook Kang, Gyujin Sim, Geonung Kim, Donguk Kim, Seungho Nam, and Sunghyun Cho. 2025. LayeringDiff: Layered Image Synthesis via Generation, then Disassembly with Generative Knowledge. *arXiv preprint arXiv:2501.01197* (2025).

Jinwoo Kim, Sangmin Han, Jinho Jeong, Jiwoo Choi, Dongyeoung Kim, and Seon Joo Kim. 2025. ORIDa: Object-centric Real-world Image Composition Dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Black Forest Labs. 2024. FLUX. https://github.com/black-forest-labs/flux.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*.

Ruibin Li, Tao Yang, Song Guo, and Lei Zhang. 2025. RORem: Training a Robust Object Remover with Human-in-the-Loop. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Lingxiao Lu, Jiangtong Li, Bo Zhang, and Li Niu. 2023a. Dreamcom: Finetuning text-guided inpainting model for image composition. *arXiv preprint arXiv:2309.15508* (2023).

Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. 2023b. Tf-icon: Diffusion-based training-free cross-domain image composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. [n. d.]. Hpsv3: Towards wide-spectrum human preference score, 2025. *arXiv preprint arxiv:2508.03789* ([n. d.]).

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. 2025. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714* (2024).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* (2022).

Jaskirat Singh, Jianming Zhang, Qing Liu, Cameron Smith, Zhe Lin, and Liang Zheng. 2024. Smartmask: Context aware high-fidelity mask generation for fine-grained object insertion and layout control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wensong Song, Hong Jiang, Zongxing Yang, Ruijie Quan, and Yi Yang. 2025. Insert anything: Image insertion via in-context editing in dit. *arXiv preprint arXiv:2504.15009* (2025).

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. 2023. Objectstitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel Aliaga. 2024. Imprint: Generative object compositing by learning identity-preserving representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gemma Canet Tarrés, Zhe Lin, Zhifei Zhang, He Zhang, Andrew Gilbert, John Collomosse, and Soo Ye Kim. 2025. Multitwine: Multi-object compositing with text and layout control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. 2024. Add-it: Training-free object insertion in images with pretrained diffusion models. *arXiv preprint arXiv:2411.07232* (2024).

Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. 2024. Mulan: A multi layer annotated dataset for controllable text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Navve Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. 2025. Paint by inpaint: Learning to add image objects by removing them first. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. 2024a. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*.

Daniel Winter, Asaf Shul, Matan Cohen, Dana Berman, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. 2024b. Objectmate: A recurrence prior for object insertion and subject-driven generation. *arXiv preprint arXiv:2412.08645* (2024).

Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. 2023. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Jinrui Yang, Qing Liu, Yijun Li, Soo Ye Kim, Daniil Pakhomov, Mengwei Ren, Jianming Zhang, Zhe Lin, Cihang Xie, and Yuyin Zhou. 2025. Generative Image Layer Decomposition with Visual Effects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. 2024. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion* (2024).

Yongsheng Yu, Ziyun Zeng, Haitian Zheng, and Jiebo Luo. 2025. Omnipaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. *arXiv preprint arXiv:2503.08677* (2025).

Jooyeol Yun, Davide Abati, Mohamed Omran, Jaegul Choo, Amirhossein Habibian, and Auke Wiggers. 2024. Generative location modeling for spatially aware object insertion. *arXiv preprint arXiv:2410.13564* (2024).

Bo Zhang, Yuxuan Duan, Jun Lan, Yan Hong, Huijia Zhu, Weiqiang Wang, and Li Niu. 2023a. Controlcom: Controllable image composition using diffusion model. *arXiv preprint arXiv:2308.10040* (2023).

Lvmin Zhang and Maneesh Agrawala. 2024. Transparent Image Layer Diffusion using Latent Transparency. *arXiv preprint arXiv:2402.17113* (2024).

Xinyang Zhang, Wentian Zhao, Xin Lu, and Jeff Chien. 2023b. Text2Layer: Layered Image Generation using Latent Diffusion Model. *arXiv preprint arXiv:2307.09781* (2023).

Jixin Zhao, Shangchen Zhou, Zhouxia Wang, Peiqing Yang, and Chen Change Loy. 2025. ObjectClear: Complete Object Removal via Object-Effect Attention. *arXiv preprint arXiv:2505.22636* (2025).

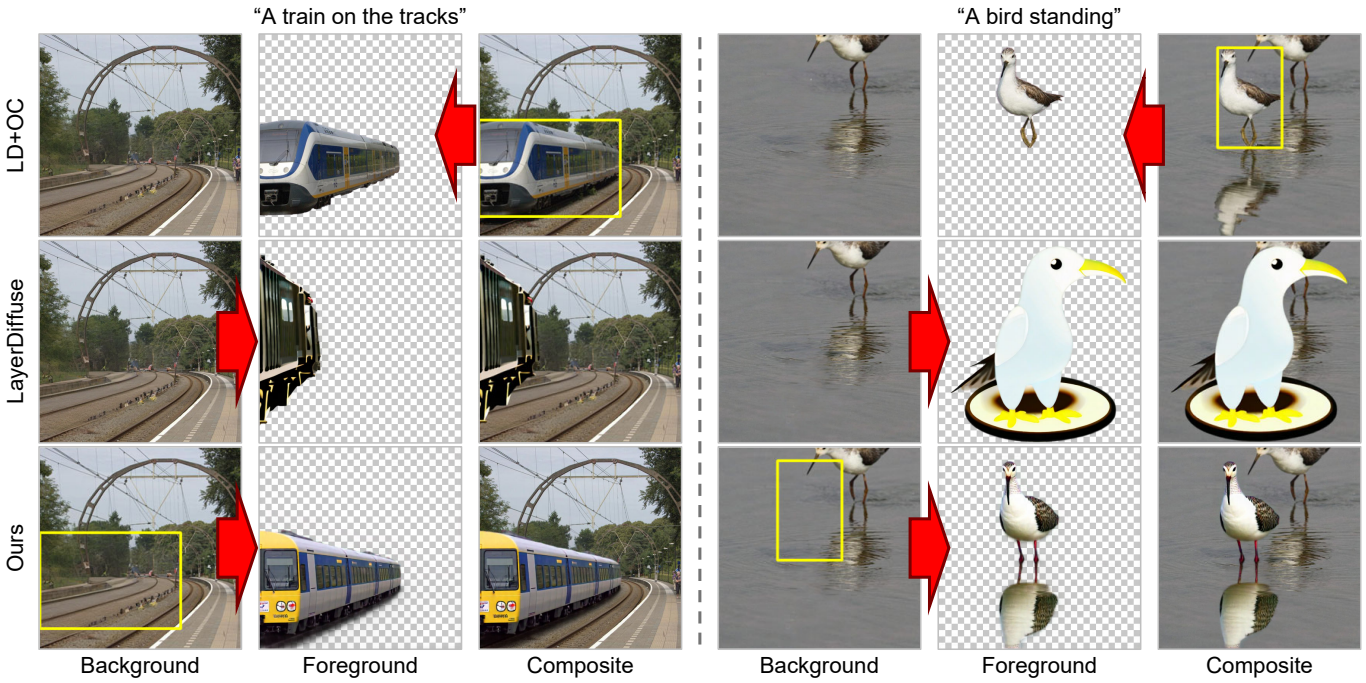

Fig. 6. Qualitative comparison with existing layered image synthesis approaches, LD+OC [Kang et al. 2025; Zhao et al. 2025] and LayerDiffuse [Zhang and Agrawala 2024]. The yellow box denotes the input mask, which is not used by LayerDiffuse.

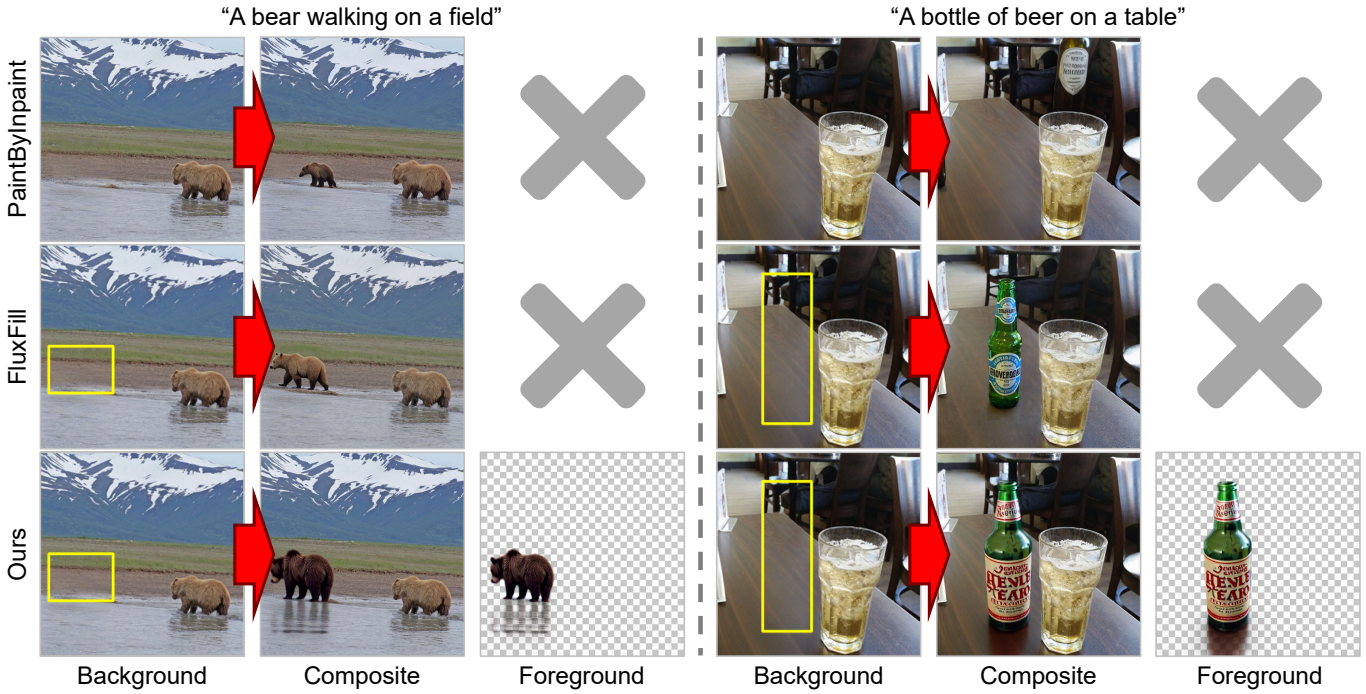

Fig. 7. Qualitative comparison with an object insertion method, PaintByInpaint [Wasserman et al. 2025], and an image inpainting method, Flux-Fill [Labs 2024]. In contrast to these methods, ours additionally produces explicit and reusable foreground layers. The yellow box denotes the input mask.

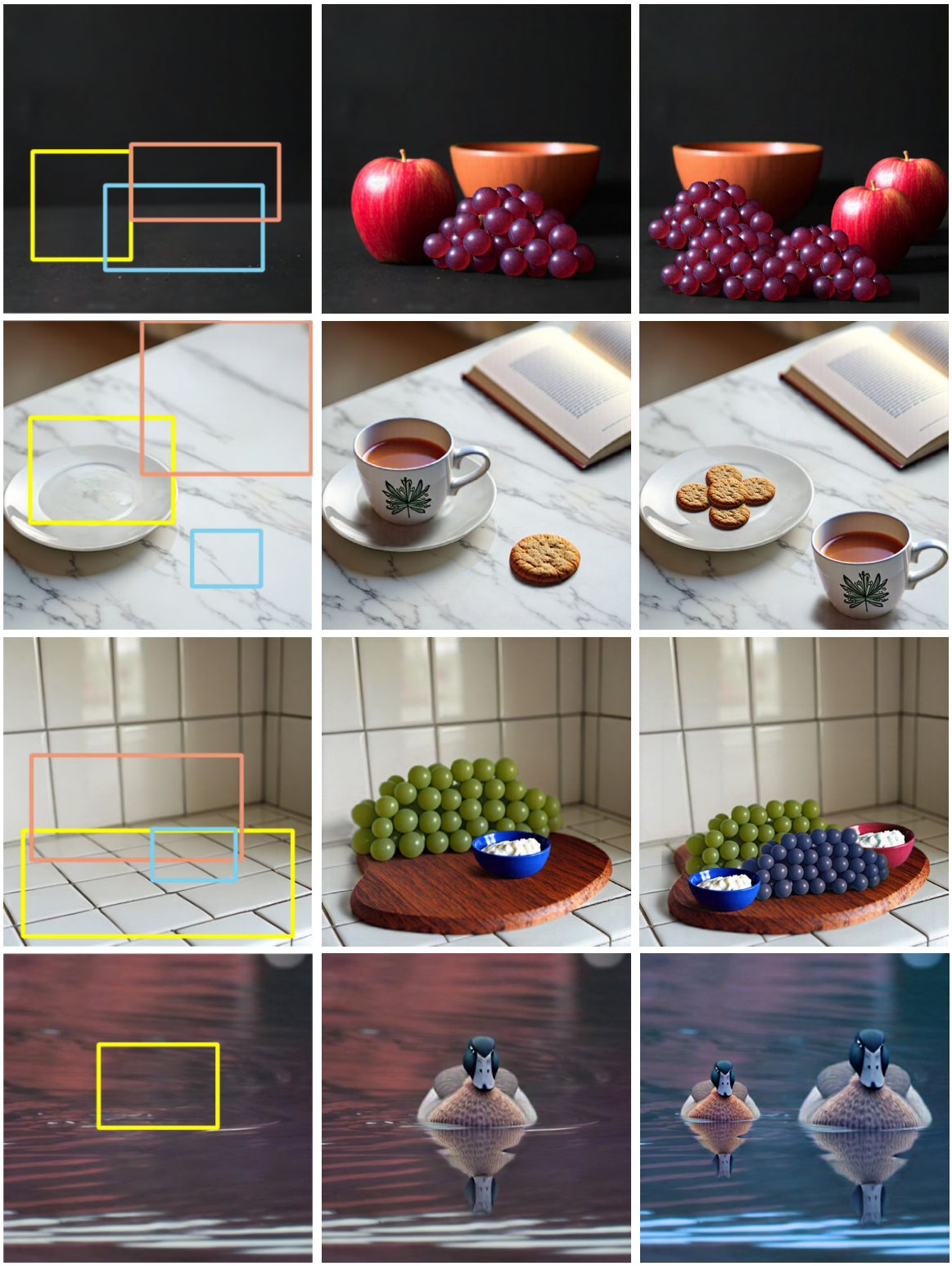

Fig. 8. Post-hoc layered image editing. Input background images (a) are first extended into richer scenes by adding diverse objects with BFS (b), and are then further manipulated through layer-wise edits (c).