RenderMatte: Exact-Alpha Rendering and Group-Relative Alignment for Image Matting

Zecheng Ren[1], **Yafei Hu**[2], **Jianing Zhao**[1], **Ruichen Cong**[1], **Qun Jin**[1]*, **Yiren Song**[3]

[1]Waseda University
[2]Shanghai Jiao Tong University
[3]National University of Singapore

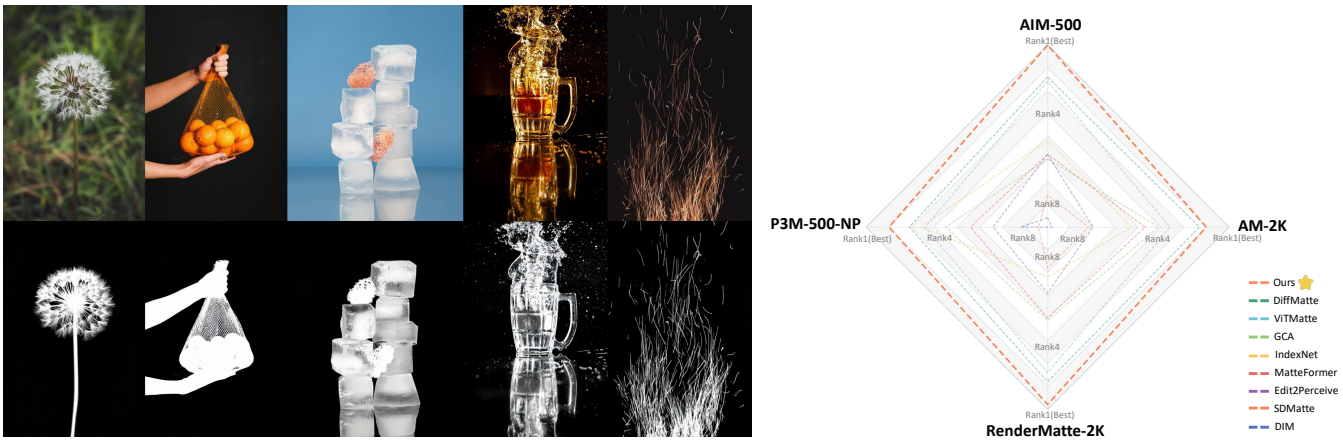


Figure 1: We present **RenderMatte**, a trimap-guided image matting framework built upon image editing priors. Our model achieves state-of-the-art performance across the zero-shot matting task, consistently outperforming previous methods. Radar chart summarizing average rank performance across four benchmarks; points closer to the outer edge indicate superior performance. **Zoom in for better view.**

Abstract

Image matting is an essential enabling technology for modern visual content production, where foreground extraction determines the realism and editability of downstream creation workflows. However, precise alpha estimation in open-world scenes remains challenging because real foregrounds exhibit highly diverse appearances and opacity patterns. This makes existing methods struggle with semantic ambiguity and fine-grained opacity variation, especially in sparse boundary regions that are fragile and difficult to supervise. To address this gap, we present **RenderMatte**, a trimap-guided matting framework that adapts FLUX.1 Kontext through full-parameter fine-tuning, leveraging image editing priors for structure-preserving alpha prediction. During supervised adaptation, an alpha-edge objective preserves the latent flow-matching signal while strengthening pixel-space boundary supervision. We further introduce group-relative alpha alignment for post-training. It compares multiple mattes sampled under the same trimap condition using matting-specific rewards for alpha accuracy, boundary fidelity, trimap compliance, and compositional consistency. To overcome the lack of precise edge annotations, we construct the **RenderMatte dataset**, a large-scale synthetic dataset combining 3D-rendered RGBA foregrounds with diverse multi-source assets. It features exact strand-level alpha annotations and diverse background composites. Experiments show state-of-the-art performance across all benchmarks, demonstrating a scalable path toward high-fidelity matting in open-world scenes.

*Corresponding Author.

Code — https://github.com/Nislab-r/RenderMatte
Dataset — https://huggingface.co/datasets/Renz-7/RenderMatte-dataset

Introduction

Image matting is an essential enabling technology for modern visual content production. By estimating a continuous per-pixel alpha matte, it separates foreground content from its background and supports realistic compositing, interactive editing, virtual production, and generative creation workflows (Levin, Lischinski, and Weiss 2006; Xu et al. 2017; Sun, Tang, and Tai 2023; Dai et al. 2025; Yin et al. 2025). Unlike semantic segmentation, matting must recover fractional opacity at mixed pixels, which makes the problem severely under-constrained. Practical systems therefore often rely on auxiliary guidance, such as trimaps, to mark definite foreground, definite background, and unknown regions where

alpha estimation is needed (Xu et al. 2017; Park et al. 2023; Ye et al. 2024; Huang et al. 2025).

However, trimap guidance does not remove the core difficulty of matting in open-world scenes. Real foregrounds exhibit highly diverse appearances and opacity patterns, while the most difficult regions are often visually small but perceptually decisive. These regions require both semantic understanding and precise low-level opacity estimation to avoid foreground-background confusion and preserve boundary details. Existing matting pipelines still struggle with this combination due to the scarcity of accurate high-frequency alpha supervision (Burgert et al. 2024; Chen et al. 2026).

Large generative models provide a promising way to strengthen the missing visual prior. They learn rich semantic and structural knowledge from large-scale image data, which can help resolve ambiguous foreground appearances (Ramesh et al. 2022; Saharia et al. 2022; Rombach et al. 2022; Podell et al. 2024; Ke et al. 2024; Hu et al. 2024). Yet matting is not a free-form generation task, since the output must stay spatially aligned with the input and represent opacity rather than RGB appearance. This makes image editing models especially relevant, as they are trained to transform an input image while preserving unrelated structures, making them better suited to structure-preserving alpha prediction (Brooks, Holynski, and Efros 2023; Zhang, Rao, and Agrawala 2023; Shi, Song, and Shou 2025).

To address these challenges, we develop **RenderMatte**, a unified framework that builds on the image-editing prior of FLUX.1 Kontext (Labs et al. 2025). Given an RGB image and a trimap, we formulate matting as a conditional image editing task and fine-tune the model to produce a pixel-aligned alpha matte instead of an RGB image. This adaptation transfers semantic and structural priors to fractional-opacity prediction. However, supervised fine-tuning may still overfit to synthetic compositing artifacts or background context rather than the actual foreground structures. To force the model to focus on foreground-intrinsic boundaries, we therefore introduce a matting-specific alpha-edge loss to enforce alpha consistency across image recomposition and repeated matting, encouraging the model to recover foreground-intrinsic opacity structures and fine details.

We further introduce group-relative alpha alignment for post-training, inspired by recent policy optimization methods for visual generation (Black et al. 2023; Wallace et al. 2024; Xue et al. 2025). Standard regression losses optimize a single prediction with absolute pixel-wise errors, so they are often dominated by large easy regions with alpha values near 0 or 1. Boundary-weighted losses reduce this imbalance, but still treat each prediction independently. In contrast, our generative model produces multiple candidate mattes for the same input. A task-specific reward compares these candidates by alpha precision, boundary quality, trimap compliance, and compositional consistency. This relative comparison encourages the model to favor predictions with better overall matting quality and finer boundary details.

Finally, we construct the **RenderMatte dataset** to address the data bottleneck behind high-fidelity matting. Existing benchmarks are too small or category-biased to capture real-world foreground diversity, while training data often lack precise supervision around sparse boundaries (Li, Zhang, and Tao 2021; Li et al. 2021, 2022; Burgert et al. 2024). To overcome this, we build a scalable synthesis pipeline centered on multi-source RGBA foregrounds. The main source is 3D-rendered foreground assets, which provide exact strand-level alpha annotations for fine structures such as hair and fur (Greff et al. 2022; Roberts et al. 2021; Deitke et al. 2022). We also include other RGBA foreground assets to broaden the foreground distribution. These foregrounds are composited with diverse natural backgrounds, producing reliable data for both supervised adaptation and reward-based post-training.

Our main contributions are summarized as follows:

- We introduce **RenderMatte**, a trimap-guided matting framework that adapts FLUX.1 Kontext through full-parameter fine-tuning, with an alpha-edge loss for foreground-intrinsic boundary preservation.
- We propose a group-relative alpha alignment strategy that treats sampled mattes as competing candidates under the same trimap condition and optimizes their relative quality using dense matting rewards.
- We construct the **RenderMatte dataset**, a multi-source synthetic matting dataset with $83{,}533$ training composites and a $2{,}000$-image test set, providing exact alpha supervision for transparent objects and complex boundaries.

Related Work

Image Matting

Classical matting methods estimate alpha mattes from low-level assumptions, such as color affinity and local sampling (Levin, Lischinski, and Weiss 2006; Chen, Li, and Tang 2013). Deep trimap-guided methods such as DIM, IndexNet, GCA, and FBA improve low-level alpha prediction through learned regression, index-guided upsampling, contextual attention, and foreground-background-alpha estimation (Xu et al. 2017; Lu et al. 2019; Hou and Liu 2019; Li and Lu 2020; Forte and Pitié 2020). Trimap-free portrait matting further reduces user guidance through objective decomposition (Ke et al. 2022). These methods directly target boundary reconstruction, but remain limited by the scale and precision of alpha supervision. To strengthen semantic reasoning, MatteFormer, ViTMatte, and unified matting frameworks introduce transformer backbones, pretrained ViTs, and stronger guidance mechanisms (Chen et al. 2018; Lin et al. 2022; Park et al. 2022; Yao et al. 2024; Guo et al. 2024; Ye et al. 2024). However, high-frequency structures such as hair, fur, and thin objects remain difficult when alpha supervision does not explicitly capture sparse boundary details. RenderMatte addresses this with exact RGBA supervision, trimap-aware alpha-edge loss, and group-relative alpha alignment.

Image Generation and Editing Models

To obtain stronger visual priors, recent dense prediction methods adapt large generative models instead of relying only on task-specific predictors. Diffusion and flow-based models learn broad visual knowledge from large-scale image data (Ho, Jain, and Abbeel 2020; Rombach et al. 2022; Lipman et al. 2022; Podell et al. 2024). These models have been

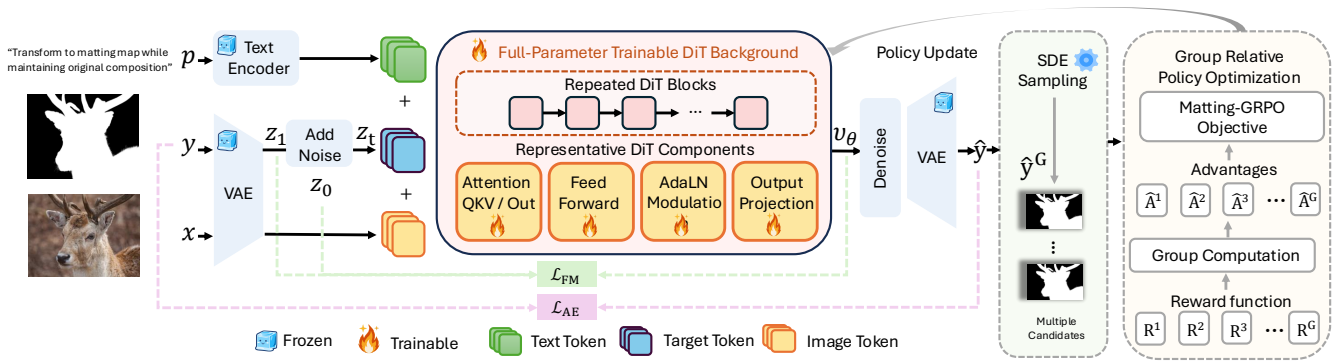


Figure 2: **Overview of the RenderMatte framework.** We adapt the FLUX.1 Kontext editor for image matting (Labs et al. 2025). Given a text prompt p, target image y, and input image x, the model predicts an alpha matte $\hat{y}$. In the forward process, the target alpha token is encoded as z_1, noised into z_t, and concatenated with the text and image tokens. The trainable DiT backbone predicts the velocity v_θ from noise z_0 to the target latent z_1. Supervised fine-tuning combines the latent-space flow matching loss $\mathcal{L}_{\mathrm{FM}}$ with the alpha-edge loss $\mathcal{L}_{\mathrm{AE}}$. We then perform group-relative alpha alignment: SDE sampling produces multiple candidate mattes $\{\hat{y}^i\}_{i=1}^G$, whose rewards $\{R^i\}_{i=1}^G$ are converted into relative advantages $\{\hat{A}^i\}_{i=1}^G$ through a GRPO-style objective.

adapted to depth estimation and image matting, including Marigold, DiffMatte, MattingGen, and SDMatte (Ke et al. 2024; Hu et al. 2024; Wang et al. 2024; Huang et al. 2025). Many of these methods start from text-to-image generators, which are optimized for concept-to-image synthesis rather than deterministic image-to-image reasoning (Shi, Song, and Shou 2025). Image editing diffusion models reduce this mismatch because they condition on an existing image and learn structure-preserving transformations (Brooks, Holynski, and Efros 2023; Zhang, Rao, and Agrawala 2023; Labs et al. 2025). FE2E and Edit2Perceive validate this editing-based paradigm for dense perception through image-to-image consistency and dense correspondence (Wang et al. 2026; Shi, Song, and Shou 2025). This motivates our use of FLUX.1 Kontext for trimap-guided alpha prediction, where the output must remain spatially aligned with the input while representing opacity rather than RGB appearance (Labs et al. 2025).

Reward-Based Post-Training for Visual Generation

Post-training aligns generative models with objectives beyond likelihood training. DDPO and DPOK formulate diffusion fine-tuning as policy optimization with reward feedback (Black et al. 2023; Fan et al. 2023). Diffusion-DPO and AlignProp instead optimize preference or differentiable rewards for text-to-image generation (Wallace et al. 2024; Prabhudesai et al. 2024). Recent GRPO-style methods further show that relative candidate quality can provide a stable signal for visual generation alignment (Liang et al. 2025; Xue et al. 2025). These methods mainly optimize global generation quality, such as aesthetics, text alignment, or preference scores. Image matting requires reference-based dense quality instead: the output must be spatially aligned, numerically precise, and sensitive to boundary errors. Different from prior GRPO-style visual generation methods that optimize global image preferences, we compare alpha mattes conditioned on the same image-trimap pair. The resulting relative advantage reflects dense alpha errors, boundary fidelity, connectivity, and trimap consistency.

Method

This section presents the full RenderMatte pipeline. The key idea is to couple exact-alpha supervision with candidate-level reward alignment: supervised adaptation teaches the model to represent alpha mattes, while group-relative post-training selects among plausible sampled mattes using matting-specific dense quality signals.

Overall Architecture

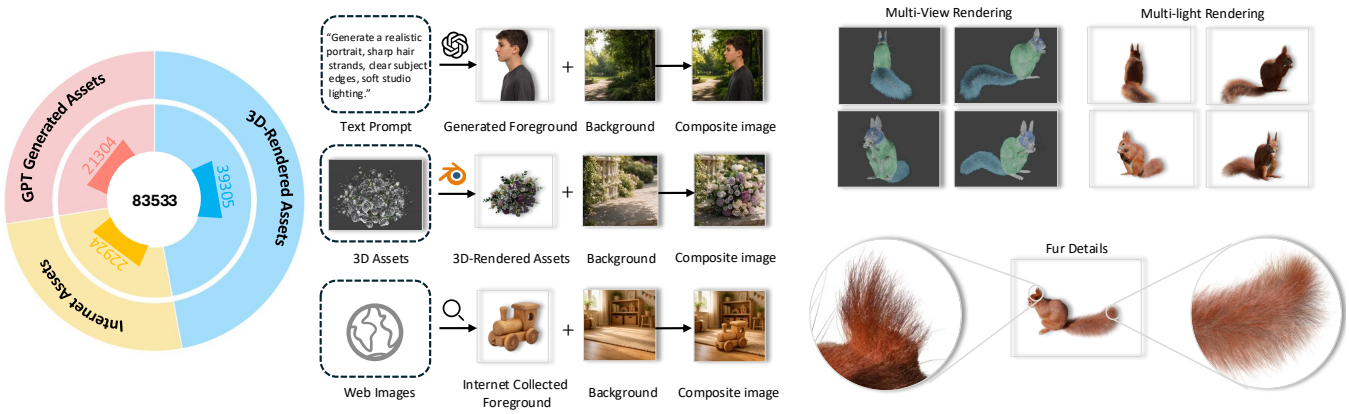
RenderMatte adapts FLUX.1 Kontext from image editing to trimap-guided alpha matting (Labs et al. 2025). Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, trimap guidance $g \in [0,1]^{H\times W}$, and a text prompt p, our goal is to predict the alpha matte $y \in [0,1]^{H\times W}$. During training, g is derived from the alpha matte y. We reshape the alpha matte target and trimap guidance into image-like inputs to match the image-editing backbone.

As shown in Fig. 2, RenderMatte contains two training stages. The first stage performs supervised adaptation, where latent flow matching transfers the editing prior to alpha prediction and the alpha-edge loss adds pixel-space boundary supervision. The second stage performs group-relative alpha alignment, where multiple candidate mattes are sampled under the same image-trimap condition and ranked by matting-specific rewards. This design separates alpha representation learning from boundary-sensitive quality alignment.

The visual condition, formed by the input image and trimap guidance, is encoded as $c_{x,g}$, while the text instruction p is encoded as c_p. During supervised adaptation, the target dense map y is projected into a target latent z_1 via the frozen VAE encoder. To perform flow matching within the DiT, a noisy latent z_t is constructed along a linear trajectory, patchified, and encoded into target tokens. These target tokens, along with the image and text tokens, are processed by the DiT backbone (Peebles and Xie 2023).

Alpha-Edge Supervised Objective

Flow-matching loss. We first adapt the generative editor to the matting domain using the rectified flow objective (Liu,



(a) Dataset Overview **(b) Multi-Source Data Synthesis Pipeline** **(c) 3D Assets Multi-View and Multi-Light Rendering**

Figure 3: **Overview of RenderMatte dataset construction.** (a) RenderMatte contains 83,533 composite training images from three foreground sources: 3D-rendered assets, GPT-generated assets, and Internet-collected assets. (b) We synthesize training samples by pairing each foreground with a background image. This pipeline preserves exact alpha annotations while increasing foreground–background diversity. (c) For 3D assets, we render each foreground under multiple camera viewpoints and lighting directions, producing diverse silhouettes, shading patterns, and strand-level hair or fur boundaries.

Gong, and Liu 2023). We sample a pure Gaussian noise latent $z_0 \sim \mathcal{N}(0, \mathbf{I})$ and construct a straight-line trajectory to define the intermediate noisy latent z_t at timestep $t \in [0, 1]$:

$$z_t = (1-t)z_0 + tz_1. \tag{1}$$

Under this formulation, the constant velocity vector of this path is non-time-dependent and simplifies to $\mathbf{v} = z_1 - z_0$. The DiT backbone $\mathbf{v}_\theta$ is trained to predict this velocity, conditioned on the intermediate state z_t, timestep t, and encoded conditions $c_{x,g}$ and c_p, by minimizing the flow-matching objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,z_1,z_0,c_{x,g},c_p}\left[\|\mathbf{v}_\theta(\text{concat}(z_t, c_{x,g}, c_p), t) - \mathbf{v}\|_2^2\right]. \tag{2}$$

Alpha-Edge loss. Latent flow matching provides stable generative adaptation, but it does not directly enforce alpha accuracy in pixel space. We therefore decode the predicted latent into an alpha matte $\hat{y}$ and introduce an alpha-edge objective that explicitly separates uncertain and certain trimap regions while adding multi-scale boundary supervision on the ambiguous band.

Let Ω_u and Ω_k denote the uncertain and certain regions defined by the trimap guidance g. The alpha-edge loss is

$$\begin{aligned}\mathcal{L}_{\text{AE}} = &\frac{1}{|\Omega_u|}\sum_{q\in\Omega_u}|\hat{y}_q - y_q| + \frac{1}{|\Omega_k|}\sum_{q\in\Omega_k}|\hat{y}_q - y_q| \\ &+ \lambda_{\text{bd}}\mathcal{L}_{\text{bd}}(\hat{y}, y; \Omega_u),\end{aligned} \tag{3}$$

where q indexes pixels, $|\Omega|$ is the region size, and λ_{bd} weights the boundary-sensitive detail term $\mathcal{L}_{\text{bd}}$, implemented as a multi-scale Laplacian pyramid loss over the uncertain region (Burt and Adelson 1987). This term encourages the model to recover high-frequency alpha structures across multiple spatial scales.

Adaptive loss weighting. We jointly optimize the latent-space and pixel-space objectives:

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{AE}}(s)\mathcal{L}_{\text{AE}}, \tag{4}$$

where s denotes the training step. Instead of using a fixed weight, we adaptively balance the two losses by their detached magnitudes:

$$\lambda_{\text{AE}}(s) = \frac{\text{sg}(\mathcal{L}_{\text{FM}})}{\text{sg}(\mathcal{L}_{\text{AE}}) + \epsilon_{\text{num}}}\gamma(s), \tag{5}$$

where $\text{sg}(\cdot)$ denotes stop-gradient and ϵ_{num} is set to 10^{-3} for numerical stability. The schedule term is implemented as $\gamma(s) = \max(0, s/N_{\text{iter}} - e_0)$, where N_{iter} is the number of iterations per epoch and e_0 is the epoch at which the pixel-space objective is enabled.

Group-Relative Alpha Alignment

GRPO-style policy objective. Although supervised adaptation provides a strong initialization, its fixed surrogate losses do not directly optimize the non-differentiable metrics used to evaluate matting quality. We therefore view the denoising trajectory that generates an alpha matte as a policy, and use the relative quality among mattes sampled from the same image-trimap condition as the optimization signal. For each conditioning context $x_c = (c_{x,g}, c_p)$, the current policy π_θ samples a group of G candidate alpha mattes $\{\hat{y}^i\}_{i=1}^G$.

Each candidate $\hat{y}^i$ is evaluated by a comprehensive matting reward function to receive a scalar score R^i. Instead of introducing a separate critic network, GRPO evaluates the trajectory-level relative advantage $\hat{A}^i$ by standardizing the rewards within the sampled group:

$$\hat{A}^i = \frac{R^i - \text{mean}(\{R^j\}_{j=1}^G)}{\text{std}(\{R^j\}_{j=1}^G) + \epsilon_{\text{std}}}, \tag{6}$$

where ϵ_{std} is a small constant for numerical stability.

This trajectory-level advantage is then broadcast to all latent transitions along the corresponding sampling path. We optimize the policy parameters over the generative flow tra-

jectory via a clipped surrogate objective:

$$\mathcal{L}_{\mathrm{GRPO}} = -\mathbb{E}_{x_c,i,t}\Big[\min\left(\rho_{i,t}\hat{A}^i, \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}^i\right)\Big], \tag{7}$$

where ϵ is the clipping threshold and the importance sampling ratio $\rho_{i,t}$ is defined as:

$$\rho_{i,t} = \exp\left(\log \pi_\theta(a_{i,t} \mid z_{i,t}, x_c) - \log \pi_{\mathrm{old}}(a_{i,t} \mid z_{i,t}, x_c)\right). \tag{8}$$

Here, π_{old} denotes the policy used to generate the sampled trajectories before the current update. $z_{i,t}$ represents the intermediate latent state at trajectory step t, and $a_{i,t}$ denotes the sampled latent transition from $z_{i,t}$ to $z_{i,t-1}$. Following generative policy optimization paradigms (Black et al. 2023; Xue et al. 2025), the network-predicted velocity parameterizes the Gaussian transition mean used to compute the log-probability of $a_{i,t}$. The single image-level reward R^i is uniformly shared across all trajectory steps for the same candidate.

Matting reward. We define the alpha-space reward to be consistent with the alpha-edge objective. Specifically, the terminal reward uses the same family of alpha-error criteria, but converts the resulting errors into a bounded score for group-relative alpha comparison. For the i-th sampled candidate, we define:

$$R^i = -\sum_{m\in\mathcal{M}_{\mathrm{R}}} w_m \min\left(\frac{e_m(\hat{y}^i, y, g)}{s_m}, \tau\right), \tag{9}$$

where $e_m(\cdot)$ is the error function for the m-th criterion in $\mathcal{M}_{\mathrm{R}}$, covering alpha accuracy, boundary fidelity, structural consistency, and trimap compliance. The negative sign converts lower errors into higher rewards; w_m, s_m, and τ denote the criterion weight, normalization scale, and clipping threshold, respectively.

Unlike generic image-level aesthetic or preference rewards, this reward is defined entirely in alpha space and explicitly depends on the trimap partition, thereby penalizing errors in the unknown boundary and violations in known foreground/background regions.

Large-Scale Synthetic Matting Data

Multi-source foreground construction. High-quality alpha supervision is difficult to obtain at scale, especially for hair, fur, thin objects, and transparent or semi-transparent regions. Existing matting datasets provide limited coverage of these challenging alpha structures, despite their importance for high-fidelity matting. Following the common compositing-based data construction paradigm in image matting (Xu et al. 2017; Li et al. 2022), we construct the RenderMatte dataset through a multi-source data synthesis pipeline, as shown in Fig. 3. The foreground asset pool contains $25{,}211$ RGBA foregrounds from three sources: 3D-rendered assets, GPT-generated foreground assets, and Internet-collected foregrounds. During asset construction, we deliberately increase the proportion of transparent and semi-transparent foregrounds, which introduce complex alpha transitions and boundary ambiguities. Each foreground is composited with sampled background images using its alpha matte to produce training samples.

We synthesize more composites from 3D-rendered assets because they provide exact high-frequency alpha annotations and controllable rendering conditions (Greff et al. 2022; Roberts et al. 2021; Deitke et al. 2022). For each 3D asset, we render foregrounds under different camera viewpoints and lighting directions, increasing the diversity of poses, silhouettes, shading patterns, and boundary appearances. This process preserves fine hair and fur geometry with strand-level alpha details. Each 3D foreground produces about eight composites on average, while other assets produce two to four composites. After filtering failed synthesis cases, RenderMatte contains $83{,}533$ training composites.

Alpha compositing and guidance generation. Given a foreground color F, alpha matte y, and background image B, we synthesize the composite image I by alpha compositing (Porter and Duff 1984). Background images are drawn from the source adopted by BG-20K (Li et al. 2022). This procedure increases foreground–background diversity while preserving exact alpha annotations.

During training, we generate trimap guidance from the ground-truth alpha matte. Specifically, we apply grayscale morphological erosion and dilation with a randomly sampled radius to obtain definite foreground, definite background, and unknown regions. The resulting trimap is encoded as 1, 0, and 0.5 for foreground, background, and unknown pixels, respectively. Randomizing the morphology radius exposes the model to unknown regions with different widths.

Experiments

Implementation details. Our model is built upon FLUX.1 Kontext (Labs et al. 2025). During supervised fine-tuning, we fully fine-tune the DiT backbone and keep the text encoders and VAE frozen. Images, trimaps, and alpha mattes are resized to 1024×1024. We optimize the model with 8-bit AdamW using a learning rate of 1×10^{-5}, weight decay 0.01, bf16 mixed precision, and DeepSpeed ZeRO-2 on 8 NVIDIA H200 GPUs. The per-device batch size is 3, giving an effective batch size of 24. The training objective combines the rectified-flow matching loss, restricted to the trimap unknown region, with the proposed alpha-edge loss.

The group-relative alignment stage is initialized from the supervised fine-tuned checkpoint. We freeze the base model and train a rank-64 LoRA adapter on the DiT modules. For each input condition, we sample $G = 8$ candidate alpha mattes using 8 denoising steps, guidance scale 1.0, and noise level 0.2. We instantiate the alignment stage with a GRPO-style clipped policy objective, using a clipping range of 1×10^{-4} and no KL penalty. The alignment adapter is optimized with 8-bit AdamW in bf16, using one inner epoch per sampling epoch. Unless otherwise specified, we use a learning rate of 2×10^{-4}, an effective training batch size of 64, and evaluate checkpoints with one denoising step and guidance scale 1.0, without test-time ensembling.

Datasets and benchmarks. We train RenderMatte on our synthesized dataset. For evaluation, we use AIM-500 (Li, Zhang, and Tao 2021), P3M-500-NP (Li et al. 2021), and AM-2K (Li et al. 2022). We also evaluate on RenderMatte-2K, a $2{,}000$-sample benchmark using completely unseen

Table 1: Quantitative comparison on AIM-500, P3M-500-NP, AM-2K, and RenderMatte-2K. Lower values are better for all metrics. AvgRank is computed over the reported metrics with tied ranks averaged. The best and second-best results are highlighted. Methods marked with † use their native interaction format.

Method	AIM-500					P3M-500-NP					AM-2K					RenderMatte-2K					AvgRank↓
	MSE↓	MAD↓	SAD↓	Grad↓	Conn↓	MSE↓	MAD↓	SAD↓	Grad↓	Conn↓	MSE↓	MAD↓	SAD↓	Grad↓	Conn↓	MSE↓	MAD↓	SAD↓	Grad↓	Conn↓	
DIM	0.017	0.031	53.1	42.3	37.9	0.007	0.014	24.5	29.2	20.6	0.013	0.024	40.5	32.7	33.3	0.019	0.032	34.0	39.1	22.1	8.6
GCA	0.021	0.051	23.4	15.9	12.1	0.012	0.038	6.2	9.8	4.5	0.007	0.031	6.3	5.5	4.8	0.004	0.008	8.4	12.1	5.0	4.9
IndexNet	0.007	0.015	25.8	16.3	17.1	0.001	0.004	7.2	9.5	5.5	0.002	0.005	9.2	8.1	7.3	0.005	0.011	11.4	16.0	7.2	5.2
MatteFormer	0.009	0.016	27.0	20.3	15.1	0.002	0.005	7.9	11.8	5.7	0.002	0.005	8.5	7.2	6.3	0.004	0.008	8.6	13.9	4.7	5.3
ViTMatte	0.004	0.011	17.8	13.7	11.0	0.001	0.004	7.1	10.6	5.0	0.001	0.005	7.7	6.2	5.5	0.001	0.004	4.5	4.8	3.0	3.1
DiffMatte	0.004	0.011	17.2	13.5	11.6	0.001	0.004	6.6	10.0	4.8	0.001	0.004	6.8	5.7	5.0	0.001	0.004	4.3	4.4	3.0	2.5
SDMatte†	0.011	0.019	31.8	26.8	17.5	0.013	0.018	32.0	20.4	20.8	0.006	0.010	17.5	13.2	10.9	0.008	0.012	12.1	20.4	3.6	7.7
Edit2Perceive†	0.006	0.017	29.1	18.2	15.7	0.003	0.011	19.4	13.2	10.2	0.004	0.012	20.4	9.6	9.9	0.005	0.011	11.5	9.2	5.0	6.4
RenderMatte	0.002	0.008	13.6	11.2	9.8	0.001	0.004	6.2	8.2	4.8	0.001	0.004	6.5	5.5	5.2	0.001	0.003	3.6	3.0	2.6	1.6

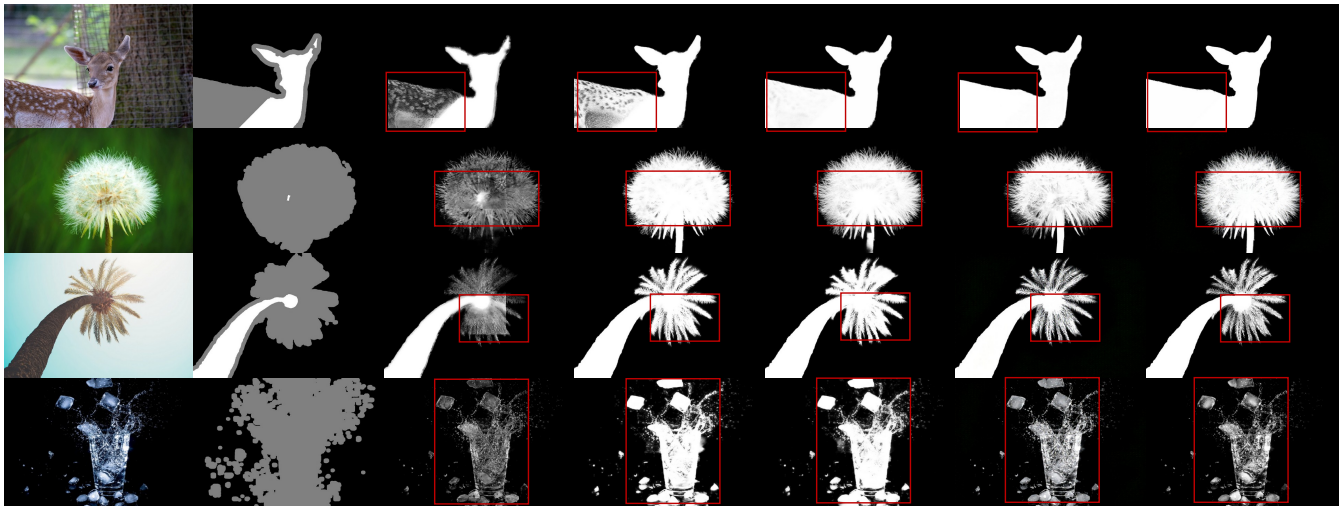


Figure 4: Qualitative comparison with state-of-the-art image matting methods. From left to right, we show the input image, trimap, predictions from competing methods, our result, and the ground-truth alpha matte. **Zoom in for better view.**

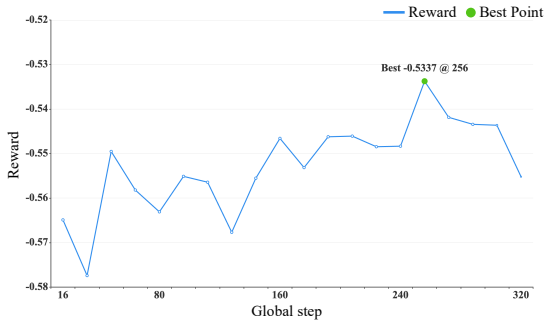


Figure 5: Evaluation reward curve of group-relative alpha alignment on AIM-500.

foregrounds and backgrounds to test zero-shot generalization.

Evaluation metrics. We adopt five standard image matting metrics: mean squared error (**MSE**), mean absolute difference (**MAD**), sum of absolute differences (**SAD**), gradient error (**Grad**), and connectivity error.

Quantitative and Qualitative Evaluation

Quantitative comparison. Table 1 compares different methods across four benchmarks. Despite the challenges of adapting generative models for precise fractional-opacity regression, RenderMatte achieves the best average rank of 1.6 and obtains the lowest errors on most evaluated metrics. The consistent reduction in Grad and SAD indicates that exact-alpha synthetic supervision, alpha-edge pixel-space refinement, and group-relative alpha alignment improve both global alpha accuracy and boundary-sensitive detail recovery. Its strong performance on public benchmarks further suggests robust generalization to diverse foreground categories and boundary structures.

We also use the reward trajectory to examine why reward-based post-training is useful beyond supervised adaptation. After supervised adaptation has saturated under differentiable losses, the reward curve in Fig. 5 continues to improve and reaches its best value at step 256. This indicates that reward-driven exploration can improve alpha predictions that are not optimized by supervised training, supporting group-relative alignment as a post-training step rather than an auxiliary supervised loss.

Qualitative comparison. Figure 4 compares the alpha mattes predicted by RenderMatte and state-of-the-art baselines. As shown in the challenging examples, existing methods often struggle with sparse or semi-transparent structures, producing coarse silhouettes, incomplete object interiors,

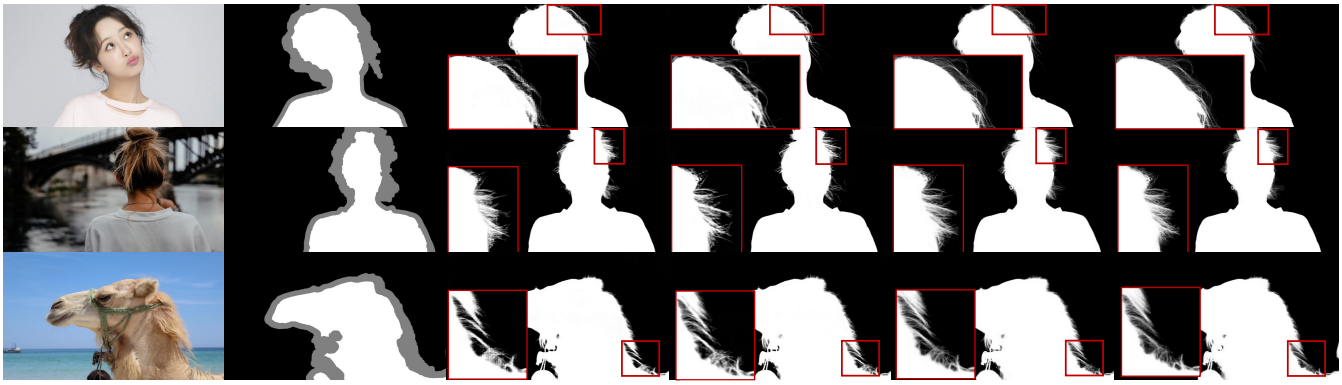


Figure 6: Visual ablation of the proposed training stages. From left to right, we show the input image, trimap, predictions from different training stages, and the ground-truth alpha matte. **Zoom in for better view.**

oversmoothed boundaries, or missing low-opacity details.

In contrast, RenderMatte preserves finer alpha structures across diverse foregrounds. It better retains animal contours, dandelion filaments, palm leaf boundaries, and transparent water splashes, while maintaining cleaner definite foreground regions. These visual results are consistent with our quantitative boundary-sensitive improvements, confirming that editing priors, alpha-edge supervision, and reward-aligned optimization jointly recover sparse boundary details.

Ablation Studies

To separate the effect of model adaptation, pixel-space alpha supervision, and reward alignment, we evaluate the same FLUX.1 Kontext backbone under three training configurations. We first examine whether adding pixel-space supervision to latent flow matching improves sparse-boundary recovery, and then evaluate whether reward-aligned post-training provides additional gains beyond supervised adaptation. Figure 6 visualizes these progressive improvements.

Effect of Alpha-Edge Supervision. As shown in Table 2, adding alpha-edge supervision brings the main improvement over the baseline and improves all matting metrics. The clearest gain appears on Grad, which directly reflects local alpha transitions and boundary sharpness. This indicates that latent flow matching alone can learn an alpha prior, but tends to smooth sparse high-frequency regions. By adding pixel-space edge supervision, the model recovers sharper contours and more stable boundaries. In Fig. 6, this effect is visible in the red boxes, where the base model misses or blurs thin boundary structures, while the alpha-edge model restores clearer silhouettes and fine strands.

Effect of Group-Relative Alpha Alignment. Group-relative alpha alignment further improves the alpha-edge model, mainly reducing SAD and Grad while also bringing a smaller gain on Conn. This suggests that reward alignment acts as a refinement stage rather than a coarse correction stage. After supervised adaptation has produced plausible alpha mattes, candidate-level reward ranking encourages the model to prefer outputs with better structural connectivity, sharper local transitions, and fewer boundary artifacts. As shown in Fig. 6, this alignment stage does not drastically change the overall foreground shape. Instead, it strengthens already recovered details in the red-box regions, especially thin hair-like structures, low-opacity edges, and fragmented boundary components. This qualitative difference explains why this reward-based refinement complements alpha-edge supervision instead of simply duplicating its effect.

Table 2: Ablation study averaged across all benchmarks. Lower values are better for all metrics. The best results are highlighted.

Base model	Alpha-Edge	RL	MSE	MAD	SAD	Grad	Conn
FLUX.1 Kontext			0.0030	0.0073	11.6	14.5	7.5
FLUX.1 Kontext	✓		0.0023	0.0063	10.1	11.6	7.0
FLUX.1 Kontext	✓	✓	0.0022	0.0061	9.6	10.7	6.7

Failure Cases and Limitations

Despite the scale of our dataset, real-world foregrounds are virtually unlimited. RenderMatte may thus struggle with rare out-of-domain objects. Inference efficiency is another limitation: processing a 1024×1024 image takes $\sim$2.7s on an A100 GPU, compared to $<$0.5s for DiffMatte. Efficient boundary-preserving inference remains future work.

Conclusion

We present RenderMatte, a trimap-guided matting framework that adapts FLUX.1 Kontext for structure-preserving alpha prediction. By combining trimap-aware alpha-edge supervision with group-relative alpha alignment, RenderMatte links exact-alpha data construction, supervised adaptation, and reward-based post-training in a unified matting pipeline. The RenderMatte dataset provides large-scale multi-source synthetic supervision for challenging transparent and fine-structure foregrounds. Experiments demonstrate state-of-the-art accuracy and boundary fidelity across standard benchmarks. We hope this framework and dataset offer a scalable path toward high-fidelity matting in open-world scenes.

References

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18392–18402.

Burgert, R. D.; Price, B. L.; Kuen, J.; Li, Y.; and Ryoo, M. S. 2024. Magick: A large-scale captioned dataset from matting generated images using chroma keying. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22595–22604.

Burt, P. J.; and Adelson, E. H. 1987. The Laplacian pyramid as a compact image code. In *Readings in computer vision*, 671–679. Elsevier.

Chen, Q.; Ge, T.; Xu, Y.; Zhang, Z.; Yang, X.; and Gai, K. 2018. Semantic human matting. In *Proceedings of the 26th ACM international conference on Multimedia*, 618–626.

Chen, Q.; Li, D.; and Tang, C.-K. 2013. KNN matting. *IEEE transactions on pattern analysis and machine intelligence*, 35(9): 2175–2188.

Chen, X.; Dong, H.; Jiang, B.; Xu, S.; Guan, Y.; Shi, K.; Gai, K.; and Song, H. 2026. αMatte4K & μMatting: Dataset and Model for Ultra-Micro Precision Alpha Video Matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12491–12500.

Dai, Y.; Li, H.; Zhou, S.; and Loy, C. C. 2025. Trans-adapter: A plug-and-play framework for transparent image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15015–15024.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2022. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*.

Fan, Y.; Watkins, O.; Du, Y.; Liu, H.; Ryu, M.; Boutilier, C.; Abbeel, P.; Ghavamzadeh, M.; Lee, K.; and Lee, K. 2023. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36: 79858–79885.

Forte, M.; and Pitié, F. 2020. F, B, Alpha Matting. *arXiv preprint arXiv:2003.07711*.

Greff, K.; Belletti, F.; Beyer, L.; Doersch, C.; Du, Y.; Duckworth, D.; Fleet, D. J.; Gnanapragasam, D.; Golemo, F.; Herrmann, C.; et al. 2022. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3749–3761.

Guo, H.; Ye, Z.; Cao, Z.; and Lu, H. 2024. In-context matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3711–3720.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hou, Q.; and Liu, F. 2019. Context-Aware Image Matting for Simultaneous Foreground and Alpha Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4130–4139.

Hu, Y.; Lin, Y.; Wang, W.; Zhao, Y.; Wei, Y.; and Shi, H. 2024. Diffusion for natural image matting. In *European Conference on Computer Vision*, 181–199. Springer.

Huang, L.; Liang, Y.; Zhang, H.; Chen, J.; Dong, W.; Chen, L.; Liu, W.; Li, B.; and Jiang, P.-T. 2025. SDMatte: Grafting Diffusion Models for Interactive Matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15229–15239.

Ke, B.; Obukhov, A.; Huang, S.; Metzger, N.; Daudt, R. C.; and Schindler, K. 2024. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9492–9502.

Ke, Z.; Sun, J.; Li, K.; Yan, Q.; and Lau, R. W. 2022. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 1140–1147.

Labs, B. F.; Batifol, S.; Blattmann, A.; Boesel, F.; Consul, S.; Diagne, C.; Dockhorn, T.; English, J.; English, Z.; Esser, P.; et al. 2025. FLUX. 1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv preprint arXiv:2506.15742*.

Levin, A.; Lischinski, D.; and Weiss, Y. 2006. A Closed Form Solution to Natural Image Matting. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, 61–68.

Li, J.; Ma, S.; Zhang, J.; and Tao, D. 2021. Privacy-preserving portrait matting. In *Proceedings of the 29th ACM international conference on multimedia*, 3501–3509.

Li, J.; Zhang, J.; Maybank, S. J.; and Tao, D. 2022. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, 130(2): 246–266.

Li, J.; Zhang, J.; and Tao, D. 2021. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*.

Li, Y.; and Lu, H. 2020. Natural image matting via guided contextual attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11450–11457.

Liang, Z.; Yuan, Y.; Gu, S.; Chen, B.; Hang, T.; Cheng, M.; Li, J.; and Zheng, L. 2025. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13199–13208.

Lin, S.; Yang, L.; Saleemi, I.; and Sengupta, S. 2022. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 238–247.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. In *The eleventh international conference on learning representations*.

Liu, X.; Gong, C.; and Liu, Q. 2023. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International conference on learning representations (ICLR)*.

Lu, H.; Dai, Y.; Shen, C.; and Xu, S. 2019. Indices matter: Learning to index for deep image matting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3266–3275.

Park, G.; Son, S.; Yoo, J.; Kim, S.; and Kwak, N. 2022. Matteformer: Transformer-based image matting via prior-tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11696–11706.

Park, K.; Woo, S.; Oh, S. W.; Kweon, I. S.; and Lee, J.-Y. 2023. Mask-guided matting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1992–2001.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2024. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, 1862–1874.

Porter, T.; and Duff, T. 1984. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 253–259.

Prabhudesai, M.; Goyal, A.; Pathak, D.; and Fragkiadaki, K. 2024. Aligning Text-to-Image Diffusion Models with Reward Backpropagation. arXiv:2310.03739.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Roberts, M.; Ramapuram, J.; Ranjan, A.; Kumar, A.; Bautista, M. A.; Paczan, N.; Webb, R.; and Susskind, J. M. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10912–10922.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*, volume 35, 36479–36494.

Shi, Y.; Song, Y.; and Shou, M. Z. 2025. Edit2Perceive: Image Editing Diffusion Models Are Strong Dense Perceivers. *arXiv preprint arXiv:2511.18673*.

Sun, Y.; Tang, C.-K.; and Tai, Y.-W. 2023. Ultrahigh resolution image/video matting with spatio-temporal sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14112–14121.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8228–8238.

Wang, J.; Lin, C.; Sun, L.; Liu, R.; Nie, L.; Li, M.; Liao, K.; and Chu, X. 2026. FE2E: From Editor to Dense Geometry Estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19844–19853.

Wang, Z.; Li, B.; Wang, J.; Liu, Y.-L.; Gu, J.; Chuang, Y.-Y.; and Satoh, S. 2024. Matting by generation. In *ACM SIGGRAPH 2024 Conference Papers*, 1–11.

Xu, N.; Price, B.; Cohen, S.; and Huang, T. 2017. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2970–2979.

Xue, Z.; Wu, J.; Gao, Y.; Kong, F.; Zhu, L.; Chen, M.; Liu, Z.; Liu, W.; Guo, Q.; Huang, W.; et al. 2025. DanceGRPO: Unleashing GRPO on Visual Generation. *arXiv preprint arXiv:2505.07818*.

Yao, J.; Wang, X.; Yang, S.; and Wang, B. 2024. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103: 102091.

Ye, Z.; Liu, W.; Guo, H.; Liang, Y.; Hong, C.; Lu, H.; and Cao, Z. 2024. Unifying automatic and interactive matting with pretrained vits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25585–25594.

Yin, S.; Zhang, Z.; Tang, Z.; Gao, K.; Xu, X.; Yan, K.; Li, J.; Chen, Y.; Chen, Y.; Shum, H.-Y.; et al. 2025. Qwen-image-layered: Towards inherent editability via layer decomposition. *arXiv preprint arXiv:2512.15603*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3836–3847.

Appendix

Overview. This appendix provides additional details that support the main paper. We first describe the construction of RenderMatte, including the foreground asset sources, foreground-background compositing statistics, and the category distribution of the RenderMatte-2K benchmark. We then report implementation details for supervised fine-tuning (SFT) and group-relative alpha alignment, including checkpoint selection, training settings, and reward-curve logging. Finally, we include representative failure cases to clarify the remaining limitations on out-of-domain examples.

Dataset Construction

RenderMatte is constructed from multi-source foreground assets and diverse background compositions. Table 3 summarizes the foreground asset sources, composited training samples, and the category distribution of RenderMatte-2K. The foreground counts correspond to unique RGBA assets from three sources: 3D-rendered assets, GPT-generated assets, and Internet-collected assets. The composited counts are therefore image-mask pairs rather than additional unique foreground categories.

The compositing ratio differs across sources. GPT-generated assets are expanded by roughly eight backgrounds per asset. 3D-rendered assets are rendered and composited more densely to increase viewpoint, lighting, pose, and boundary diversity. Internet-collected assets are composited more selectively because the source pool is already larger and more visually diverse. This design yields 25,211 foreground assets and 83,533 training composites.

We also construct RenderMatte-2K as a held-out benchmark with 2,000 samples from unseen foregrounds and backgrounds. Its category distribution covers people, animals, furniture, toys, tools, sports objects, accessories, fruits, musical objects, textiles, crafts, transparent objects, trees, and flowers.

Implementation Details

Model architecture. RenderMatte uses FLUX.1 Kontext as an image-editing backbone. The text instruction is fixed across all samples, while the visual condition is provided by the original image and the trimap. The target image is the ground-truth alpha matte represented in an image-compatible format.

Supervised Fine-Tuning. The main paper reports the core supervised fine-tuning configuration. Here we provide additional implementation details that are not included in the main text.

Task formatting. Each training sample is formatted as a FLUX Kontext image-editing instance with a fixed instruction, "Transform to matting map while maintaining original composition." The visual condition contains two reference images: the original RGB image and the trimap. The generation target is the ground-truth alpha matte. We use a fixed instruction rather than per-image captions, so the adaptation is driven by visual matting conditions rather than semantic text variation.

Online trimap construction. Trimaps are generated on the fly from the ground-truth alpha matte during training. We randomly sample the morphological kernel size from $[5, 15)$ and the number of erosion/dilation iterations from $[5, 15)$. The unknown region is defined as the pixels that remain outside both the eroded definite foreground and the dilated definite background. This online procedure exposes the model to different uncertain-band widths for the same underlying alpha matte.

Optimization schedule and checkpointing. The learning-rate schedule uses a short constant-scheduler warmup: the PyTorch `ConstantLR` default factor $1/3$ is applied for the first five steps, after which the learning rate remains constant. Gradient checkpointing is enabled, and EMA is not used in the SFT stage. Checkpoints are saved every 2,000 steps. During training, a 10-sample AIM-500 quick evaluation is run every 500 steps, and the full 500-sample AIM-500 evaluation is run at each saved checkpoint.

SFT checkpoint selection. The SFT run saved full-parameter checkpoints at steps 2,000, 4,000, 6,000, 8,000, 10,000, and 12,000. Table 4 compares the checkpoints with complete benchmark metrics. The step-10,000 checkpoint is used as the initialization for the group-relative alpha alignment stage. Although step 12,000 slightly improves AIM-500 MSE/MAD/SAD/Conn, it is flat or mildly worse on P3M-500-NP and AM-2K, especially for boundary-sensitive metrics. We therefore use step 10,000 as the more stable checkpoint for the subsequent alignment stage.

Group-Relative Alpha Alignment

The alignment stage starts from the SFT checkpoint at step 10,000. The base model is frozen, and only a rank-64 LoRA adapter on the DiT modules is trained. For each prompt-image condition, the sampler generates eight candidate mattes, forming the group used for relative advantage normalization. Training uses eight denoising steps, while evaluation uses one denoising step, both with guidance scale 1.0 and noise level 0.2. We do not use a KL penalty against the SFT reference policy in this stage.

The alignment run uses a constant learning rate without a scheduler. We enable activation checkpointing and per-prompt reward-statistic tracking. The SDE exploration is applied within a two-step window inside the eight-step sampling trajectory, which preserves stochastic exploration while limiting variance in long trajectories. Gradients are clipped with a maximum norm of 1.0, and EMA is maintained for the LoRA adapter with decay 0.9 and an update interval of 8 steps.

Reward-Curve Logging

Reward-Curve Protocol. The reward curve in the main paper is plotted from evaluation checkpoints under the same evaluation protocol, rather than from raw training summaries. In the inspected run directory, `reward_history.jsonl` contains 43 training summaries and 21 evaluation summaries. Training summaries are recorded every 8 global steps over 128 sampled rollouts, while evaluation summaries are

Table 3: Dataset statistics of RenderMatte.

Foreground asset counts by source and category.

Source	Person	Animal	Flower/Grass	Tree	Toy	Furniture	Transparent	Other	Total
GPT-generated	275	663	400	0	225	175	0	925	2,663
Internet-collected	3,479	4,459	3,002	737	61	5,701	457	0	17,896
3D-rendered	476	243	1,607	1,045	296	985	0	0	4,652
Total	4,230	5,365	5,009	1,782	582	6,861	457	925	25,211

Composited training samples after pairing foreground assets with backgrounds.

Source	Person	Animal	Flower/Grass	Tree	Toy	Furniture	Transparent	Other	Total
GPT-generated	2,200	5,304	3,200	0	1,800	1,400	0	7,400	21,304
Internet-collected	3,600	6,000	2,998	1,000	150	6,000	3,176	0	22,924
3D-rendered	6,440	8,000	6,340	9,570	6,780	2,175	0	0	39,305
Total	12,240	19,304	12,538	10,570	8,730	9,575	3,176	7,400	83,533

Category distribution of the RenderMatte-2K benchmark.

Category	People	Animal	Furniture	Toy	Tool	Sports	Accessory	Fruit	Musical	Textile	Craft	Transparent	Tree	Flower	Total
Count	220	220	170	150	140	140	130	130	130	130	120	110	110	100	2,000

Table 4: SFT checkpoint comparison for checkpoints with complete benchmark metrics reported in the training record. Lower values are better for all metrics. Step 10,000 is used to initialize the alignment stage because it gives the more balanced validation behavior across benchmarks.

Benchmark	Step	MSE	MAD	SAD	Grad	Conn
P3M-500-NP	10,000	0.0010	0.0050	8.6069	8.6598	4.9296
P3M-500-NP	12,000	0.0011	0.0051	8.7953	9.3016	5.0567
AM-2K	10,000	0.0010	0.0056	9.6483	5.6652	5.7563
AM-2K	12,000	0.0010	0.0056	9.6894	5.8540	5.7362
AIM-500	10,000	0.0025	0.0096	16.1219	11.3611	10.0452
AIM-500	12,000	0.0023	0.0093	15.7202	11.5469	9.7252

recorded every 16 global steps over 500 evaluation samples. LoRA checkpoints are saved every 16 steps from step 16 to step 320, and debug visualizations are stored for qualitative inspection. Table 5 reports the evaluation points used by the main reward curve through step 320.

Failure Cases

Figure 7 shows representative failure cases on out-of-domain data. These examples complement the limitation discussion in the main paper. Although RenderMatte benefits from exact-alpha synthetic supervision and reward-based alignment, it can still produce suboptimal mattes for rare foreground objects, unusual transparency patterns, or extreme structures that are underrepresented in the training distribution. Such cases suggest that broader foreground coverage and more diverse boundary/opacity patterns remain useful directions for future data construction.

Table 5: Evaluation checkpoints available for the reward curve through step 320. Higher reward and lower error are better.

Step	Reward	Error	MAD	SAD	Grad	Conn
16	-0.565	0.565	0.0167	4.371	3.721	2.241
32	-0.577	0.577	0.0172	4.518	3.763	2.281
48	-0.550	0.550	0.0164	4.304	3.700	2.232
64	-0.558	0.558	0.0164	4.301	3.639	2.226
80	-0.563	0.563	0.0167	4.385	3.731	2.249
96	-0.555	0.555	0.0165	4.332	3.668	2.249
112	-0.556	0.556	0.0163	4.272	3.715	2.213
128	-0.568	0.568	0.0168	4.405	3.662	2.266
144	-0.556	0.556	0.0165	4.322	3.679	2.234
160	-0.547	0.547	0.0162	4.251	3.635	2.214
176	-0.553	0.553	0.0163	4.265	3.642	2.256
192	-0.546	0.546	0.0163	4.262	3.578	2.242
208	-0.546	0.546	0.0160	4.191	3.605	2.211
224	-0.548	0.548	0.0164	4.311	3.622	2.271
240	-0.548	0.548	0.0163	4.281	3.612	2.244
256	-0.534	0.534	0.0158	4.150	3.550	2.225
272	-0.542	0.542	0.0163	4.273	3.633	2.278
288	-0.543	0.543	0.0163	4.277	3.621	2.253
304	-0.544	0.544	0.0162	4.243	3.585	2.263
320	-0.555	0.555	0.0166	4.339	3.660	2.267

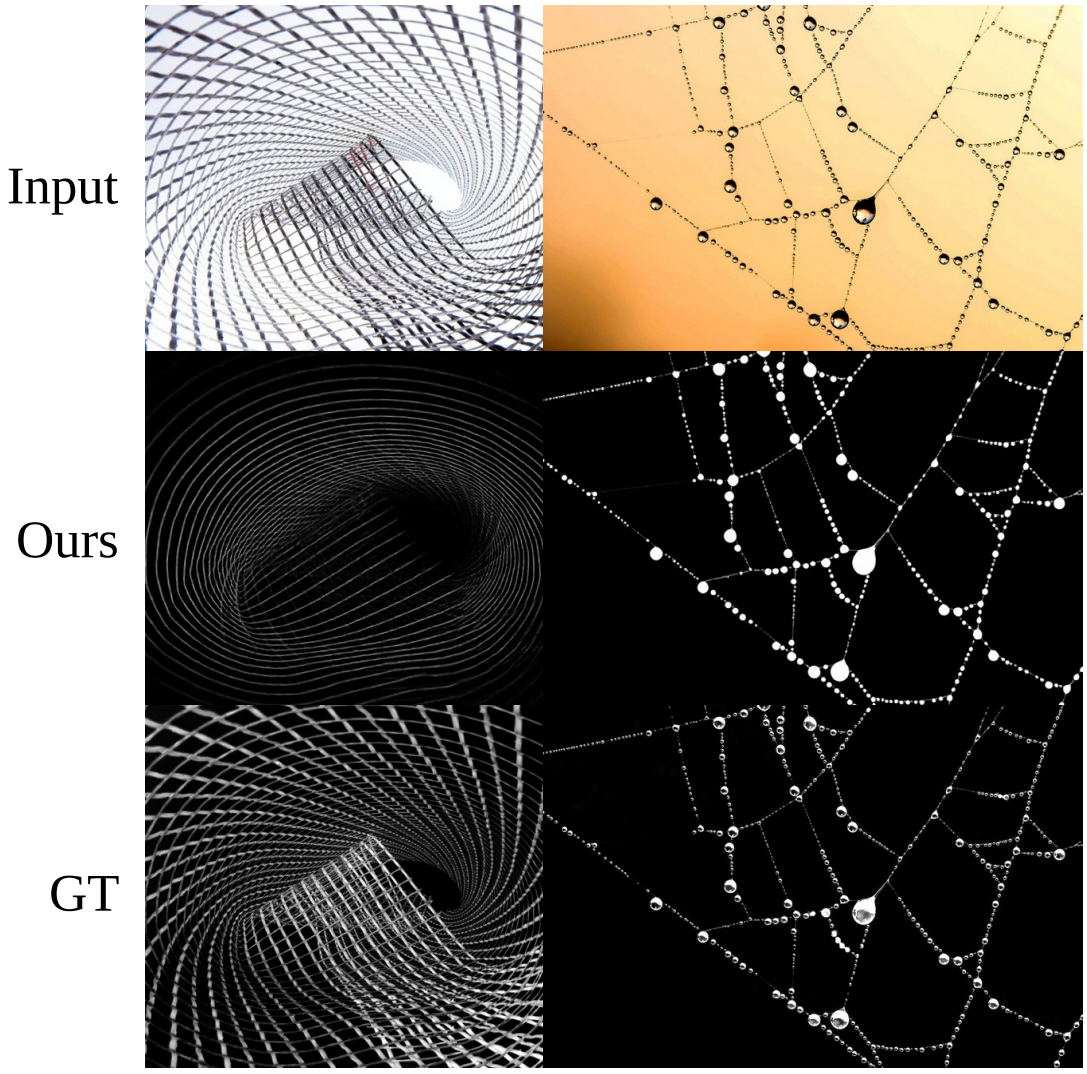


Figure 7: Failure cases on out-of-domain data. Top to bottom: input, RenderMatte, and ground truth. The model may produce suboptimal mattes for rare foreground objects or extreme structures that deviate from the training distribution.