Teaching Diffusion Models to Ground Alpha Matte

Anonymous authors
Paper under double-blind review

Abstract

The power of visual language models is showcased in visual understanding tasks, where language-guided models achieve impressive flexibility and precision. In this paper, we extend this capability to the challenging domain of image matting by framing it as a soft grounding problem, enabling a single diffusion model to handle diverse objects, textures, and transparencies, all directed by descriptive text prompts. Our method teaches the diffusion model to ground alpha mattes by guiding it through a process of instance-level localization and transparency estimation. First, we introduce an intermediate objective that trains the model to accurately localize semantic components of the matte based on natural language cues, establishing a robust spatial foundation. Building on this, the model progressively refines its transparency estimation abilities, using the learned semantic structure as a prior to enhance the precision of alpha matte predictions. By treating spatial localization and transparency estimation as distinct learning objectives, our approach allows the model to fully leverage the semantic depth of diffusion models, removing the need for rigid visual priors. Extensive experiments highlight our model's adaptability, precision, and computational efficiency, setting a new benchmark for flexible, text-driven image matting solutions.

1 Introduction

Image matting is a longstanding and foundational task in computer vision, aimed at extracting a foreground object from an image and estimating the transparency of each pixel. Traditionally, this process is modeled by the following equation (Porter & Duff, 1984):

$$I = \alpha F + (1 - \alpha)B, \tag{1}$$

where only the input image I is known, while the alpha matte α, foreground F, and background B colors are unknowns. Solving this ill-posed problem has led researchers to develop various priors, including trimaps (Levin et al., 2007; Xu et al., 2017; Yao et al., 2024a), background estimates (Lin et al., 2021), binary masks (Yu et al., 2021; Huynh et al., 2024), and user-provided inputs (Ye et al., 2024; Li et al., 2024b).

However, these visual-level priors present limitations in many matting scenarios. For instance, trimaps require substantial annotation efforts, while the background and binary masks are unsuitable for dynamic scenes without temporal information. Additionally, interactive inputs are generally limited to static settings and lack generalizability for complex textures in natural image matting (see the upper-right example in Fig. 1). Thus, there is a strong need for a unified and user-friendly approach.

Recently, visual-language approaches have achieved significant progress in dense visual prediction tasks (Gavrilyuk et al., 2018; Ye et al., 2019; Luo et al., 2020; Wu et al., 2022). Notably, text-to-image diffusion models like Stable Diffusion (SD) (Rombach et al., 2022) have enabled tasks such as open-vocabulary panoptic segmentation (Xu et al., 2023a) and referring image segmentation (Zhao et al., 2023a) by leveraging capabilities for semantic differentiation and cross-modal attention (illustrated in Fig. 2). These features make SD a promising candidate for matte grounding tasks through natural language guidance, combining the expressive and intuitive nature of language with the strong prior knowledge embedded in the SD model.

This raises a natural question: *can this prior knowledge be applied effectively to alpha matting?* Unlike standard visual grounding tasks that typically produce binary masks for object identification, alpha matting

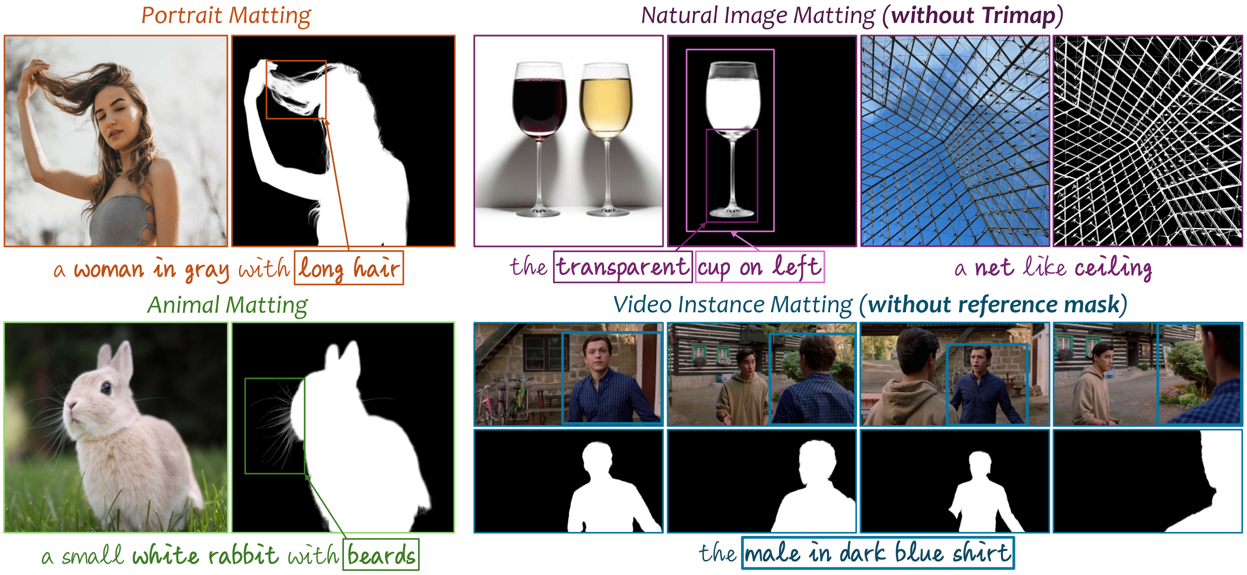

Portrait Matting
a woman in gray with long hair
Natural Image Matting (without Trimap)
the transparent cup on left
a net like ceiling
Animal Matting
a small white rabbit with beards
Video Instance Matting (without reference mask)
the male in dark blue shirt

Figure 1: We propose to teach a single diffusion model capable of handling various matting tasks using task-specific text guidance (shown below each sample) with keywords marked in **bold**. By introducing language-driven priors, we unify diverse matting tasks into a soft grounding framework. Our novel pipeline for teaching the text-to-image diffusion model achieves state-of-the-art performance on this challenging problem.

requires both precise object localization and fine-grained transparency estimation, with the alpha value $\alpha \in [0, 1]$ in Eq. (1) accurately predicted. We define this unified approach as **soft grounding**, addressing both instance-level localization and transparency simultaneously for more adaptable and nuanced matting applications.

While SD holds significant potential, directly adapting it for soft grounding tasks introduces challenges due to the simultaneous requirements of localization and transparency estimation. Previous adaptations of SD for portrait matting (Xu et al., 2024a; Wang et al., 2024) rely on the assumption of a prominent, easily distinguishable foreground object, which simplifies localization. However, this assumption is often invalid in more complex scenarios involving multiple objects or intricate transparency patterns, such as those encountered in natural image matting (Fig. 1), which complicate the separation between foreground and background. Therefore, directly applying SD may yield suboptimal results due to the increased difficulty in resolving both localization and transparency.

In this work, we propose a framework to teach SD to ground alpha mattes with any user prompts. Rather than attempting to estimate the alpha matte directly, our approach introduces an intermediate objective that initially guides the model to localize semantic components of the alpha channel. Once this "teacher" model is trained for localization, we introduce a distillation process where a "student" model leverages the teacher's learned semantic information to progressively refine transparency estimation. This sequential framework ultimately enables accurate alpha matte prediction with minimal post-processing. This approach offers two primary advantages. First, by avoiding continuous fine-tuning of the teacher model for transparency estimation, we preserve much of its pretrained semantic knowledge, maintaining a clear distinction between localization and transparency tasks. Second, the strong semantic foundation established by the teacher model enables us to employ a more computationally efficient student model, enhancing the practicality of this approach for real-world applications.

Our main contributions are as follows:

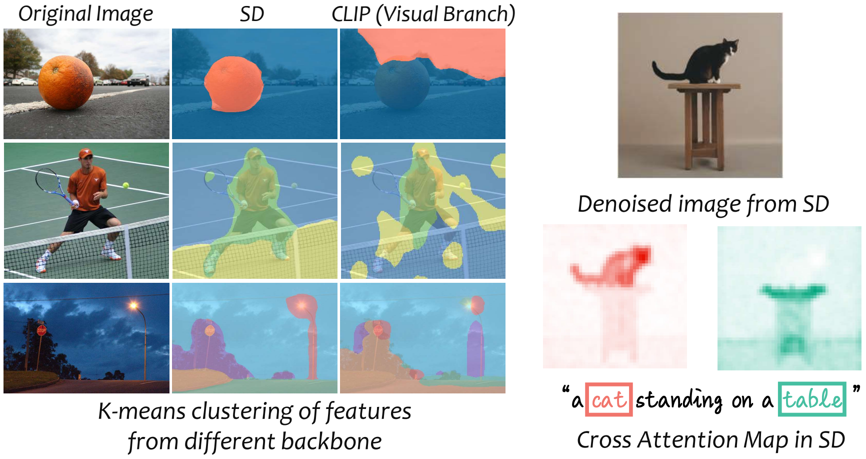

Original Image
SD
CLIP (Visual Branch)
K-means clustering of features from different backbone
Denoised image from SD
“a cat standing on a table”
Cross Attention Map in SD

Figure 2: Visualization of features from SD (Rombach et al., 2022) and CLIP (Radford et al., 2021) (left) and cross-attention maps in SD (right). SD exhibits strong semantic differentiation in its features and cross-modal attention maps, with both inter-correlation and intra-consistency.

- We unify various matting tasks into a soft grounding problem, leveraging the Stable Diffusion model's visual-language capabilities.
- We introduce a distillation framework that teaches SD to ground alpha mattes by disentangling instance-level localization and transparency estimation, enhancing performance in scenarios involving multiple objects or complex transparency.
- This framework enables optimization of the student model's structure, improving computational efficiency for practical applications.
- Extensive evaluations demonstrate that our method outperforms multiple baselines in soft grounding tasks, achieving competitive speed and generalizing well across different matting categories.

2 Related Work

2.1 Alpha Matte Grounding

Due to the inherent ambiguity in image matting, most existing methods rely heavily on trimaps as prior guidance (Levin et al., 2007; He et al., 2011; Chen et al., 2013; Xu et al., 2017; Hou & Liu, 2019; Park et al., 2022; Wang et al., 2023b; Yao et al., 2024a; Xu et al., 2023b; Hu et al., 2024). However, generating accurate trimaps that can distinguish foreground, background, and unknown regions is costly and time-intensive. Consequently, recent research has focused on exploring alternative priors, including background information (Lin et al., 2021; Sengupta et al., 2020), binary masks (Park et al., 2023; Yu et al., 2021; Huynh et al., 2024; Sun et al., 2022; Li et al., 2024a; Yang et al., 2025), in-context priors (Guo et al., 2024), and interactive inputs such as points, bounding boxes, and scribbles (Wei et al., 2021; Yang et al., 2022; Yao et al., 2024b; Ye et al., 2024; Li et al., 2024b; Xia et al., 2024).

Text-based priors, which offer more flexible and intuitive guidance by dynamically and autonomously identifying matting objects, have also been investigated. For instance, Li *et al.* (Li et al., 2023a) introduce CLIPMat, which is the first text-based matting method leveraging a pre-trained CLIP model (Radford et al., 2021) to integrate visual and textual features for matting. Xu *et al.* (Xu et al., 2024b) further enhance the feature fusion approach based on Li et al. (2023a). However, since CLIP was designed for cross-modal similarity across entire images, it struggles with the high semantic precision and detailed requirements of pixel-level matting. In contrast, our method is based on a text-to-image diffusion model that provides pixel-level visual-language priors and robust semantic differentiation, making it more suited to the nuanced demands of image matting.

2.2 Diffusion Models for Image Matting

Diffusion models have shown great potential in a variety of applications, including generative tasks (Rahman et al., 2023; Shi et al., 2024; Blattmann et al., 2023; Brooks et al., 2023; Ruiz et al., 2023; Wang et al., 2023a) and dense prediction tasks (Ji et al., 2023; Zhao et al., 2023a; Xu et al., 2023a; Lee et al., 2024; Xu et al., 2024a; Burgert et al., 2023). Some studies have applied the prior knowledge encoded in these models to image matting tasks.

Guo *et al.* (Guo et al., 2024) leverage the in-context correspondence priors within Stable Diffusion (Rombach et al., 2022), using guidance from a reference image to perform matting of the same object across different scenes. Wang *et al.* (Wang et al., 2024) approach matting as a generative task, applying denoising over multiple time steps and fine-tuning the model to predict the alpha matte. However, these methods depend on forecasting alpha mattes directly, limiting their effectiveness to matting a single object and often resulting in inefficiencies due to the high computational cost of diffusion models.

Alternatively, some methods operate directly in pixel space by diffusing a disturbed trimap (Xu et al., 2023b) or pure noise (Hu et al., 2024) until a clean alpha matte is produced. Li *et al.* (Li et al., 2024c) extend this approach to latent space, introducing modified self-attention to better model matting context. However, these approaches rely heavily on a trimap as conditioning input, limiting their flexibility in practical applications.

3 Method

3.1 Overview

Problem Formulation. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$ and a foreground text expression $\mathcal{T}$, our goal is to teach the Stable Diffusion (SD) model soft grounding and predict the alpha matte $\alpha \in [0,1]^{H \times W \times 1}$ in a single step.

Network Design. Our framework is illustrated in Fig. 3. To teach the SD model to ground the alpha matte, we disentangle the soft grounding problem into two sub-objectives using an intermediate objective, soft semantic grounding, along with a distillation framework. First, the original diffusion model is trained to localize the semantic components of the target alpha matte (see Sec. 3.2.1). Then, an asymmetric distillation framework with two tailored objectives guides the model to refine transparency estimation while preserving the localization capability learned in the initial stage (see Sec. 3.2.2). This approach ensures effective task disentanglement. Moreover, the distillation framework enables the adoption of a computationally efficient model for soft grounding, improving its practicality (see Sec. 3.3). Using the soft grounding results and output features from the well-taught diffusion model, the matting decoder can readily predict the final full-resolution alpha matte (see Sec. 3.4). We discuss our proposed paradigm below.

3.2 Soft Grounding Teaching

3.2.1 Soft Grounding in Semantic

We first teach the diffusion model to address an intermediate task: localizing each semantic component of the target alpha matte. This is achieved by fine-tuning the denoising U-Net (ϵ_θ) starting from the original weights of SD. Specifically, taking the latent code $z_I := \mathcal{E}(I)$ at a resolution of $\frac{H}{8} \times \frac{W}{8}$ and the corresponding linguistic features $f_\mathcal{T}$ as conditional input, ϵ_θ is fine-tuned to distinguish the semantic regions of alpha matte (the foreground, background, and transparent regions) and to predict classification results for each region. The results are represented as $s_I \in \{0,1\}^{\frac{H}{8} \times \frac{W}{8} \times 3}$ of I. To achieve this, we replace the last latent prediction head in the original ϵ_θ with a new head comprising two groups of ConvBNReLU layers to predict s_I. The linguistic features $f_\mathcal{T}$ are generated by the frozen CLIP text encoder (Radford et al., 2021) and further refined using a two-layer MLP text adapter, inspired by Gao et al. (2024). We train the model to predict classification probabilities $p_I \in [0,1]^{\frac{H}{8} \times \frac{W}{8} \times 3}$ for each channel in s_I by minimizing the cross-entropy loss $\mathcal{L}_{CE}$:

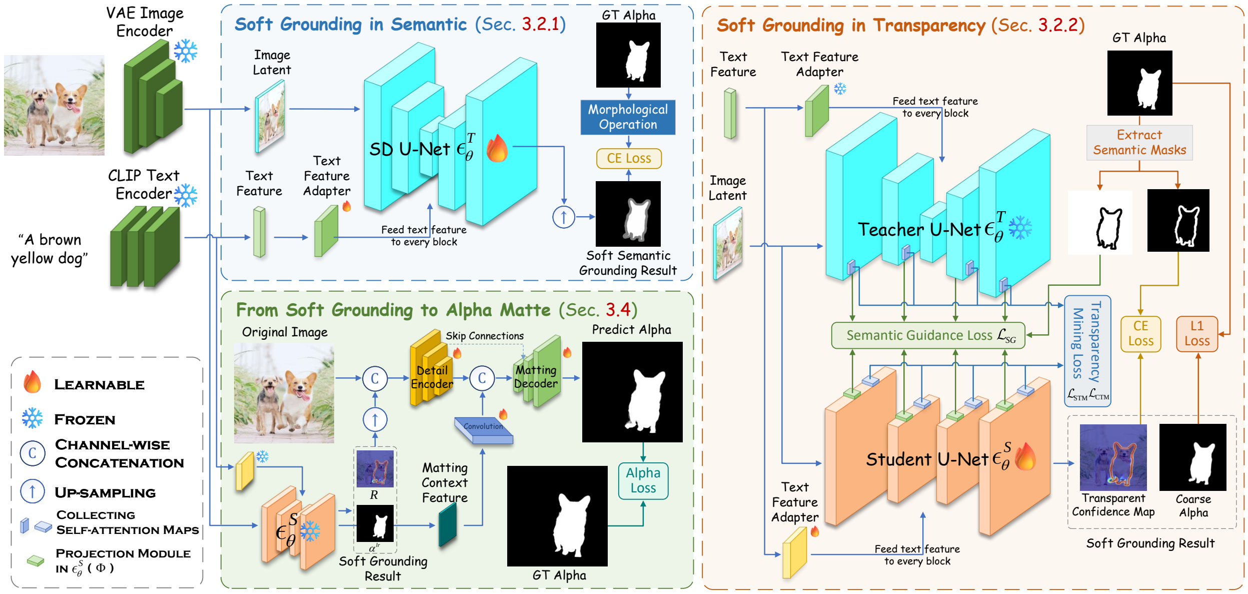

VAE Image Encoder
CLIP Text Encoder
"A brown yellow dog"
Soft Grounding in Semantic (Sec. 3.2.1)
Image Latent
Text Feature
Text Feature Adapter
SD U-Net ϵ_θ^T
Feed text feature to every block
GT Alpha
Morphological Operation
CE Loss
Soft Semantic Grounding Result
From Soft Grounding to Alpha Matte (Sec. 3.4)
Original Image
Skip Connections
Detail Encoder
Matting Decoder
Predict Alpha
Convolution
R
α^{lr}
ϵ_θ^S
Soft Grounding Result
Matting Context Feature
Alpha Loss
GT Alpha
Learnable
Frozen
C Channel-wise Concatenation
Up-sampling
Collecting Self-attention Maps
Projection Module in ϵ_θ^S (Φ)
Soft Grounding in Transparency (Sec. 3.2.2)
Text Feature
Text Feature Adapter
Feed text feature to every block
Image Latent
Teacher U-Net ϵ_θ^T
GT Alpha
Extract Semantic Masks
Semantic Guidance Loss $\mathcal{L}_{SG}$
Transparency Mining Loss $\mathcal{L}_{STM}$ $\mathcal{L}_{CTM}$
CE Loss
L1 Loss
Student U-Net ϵ_θ^S
Text Feature Adapter
Feed text feature to every block
Transparent Confidence Map
Coarse Alpha
Soft Grounding Result

Figure 3: Overview of the pipeline of our model, with unimportant operations and blocks omitted for clearer illustration. Motivated by the goal of disentangling two learning objectives, we first adapt the diffusion model to a sub-objective of instance-wise localization of semantic components of the alpha matte (blue part on top-left). Next, we propose a distillation framework (orange part on the right) to distill the localization information from the teacher model (ϵ_θ^T) into the student model (ϵ_θ^S) using the Semantic Guidance Loss ($\mathcal{L}_{\text{SG}}$) based on internel features, while encourage ϵ_θ^S to simultaneously explore the transparent-related information via the Transparency Mining Loss ($\mathcal{L}_{\text{STM}}$, $\mathcal{L}_{\text{CTM}}$) based on attention maps. ϵ_θ^S learns to predict the coarse alpha matte α^{lr} and the transparent confidence map R to further enhance α^{lr} into α at full resolution (green part on bottom-left).

$$\mathcal{L}_{\text{CE}} = -\sum_{c=1}^{3} s_{GT}^c \log(p_I^c), \tag{2}$$

where s_{GT} represents the GT region representation, obtained by applying morphological operations to the GT alpha, and p_I^c denotes the classification probability for the c-th channel in s_I. We denote the fine-tuned ϵ_θ as ϵ_θ^T.

3.2.2 Soft Grounding in Transparency

After the convergence of training ϵ_θ^T, our next goal is to further explore transparency information based on ϵ_θ^T until achieving soft grounding. A straightforward approach would be to fine-tune ϵ_θ^T to predict transparency directly. However, this presents two key challenges. First, fine-tuning ϵ_θ^T for transparency estimation risks disrupting its well-learned semantic knowledge. Second, the computational cost of ϵ_θ^T is typically high, limiting its practicality for soft grounding. To address both challenges, we propose a novel soft grounding distillation framework. This framework fully exploits the strong semantic guidance embedded in the intermediate features of the pre-trained ϵ_θ^T while enabling the student model (ϵ_θ^S) to refine transparency estimation in the corresponding alpha matte. As a result, ϵ_θ^S is expected to gain full capabilities for solving the soft grounding problem. For implementation, ϵ_θ^S is trained to predict a coarse alpha matte $\alpha^{lr} \in [0,1]^{\frac{H}{8} \times \frac{W}{8} \times 1}$ and a transparent confidence map $R \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 1}$, which indicates regions that may be transparent. These outputs are then used for further upsampling and refining α^{lr} to α. Next, we introduce two key losses to effectively guide this distillation process.

Strong Semantic Guidance Loss. Inspired by Kim et al. (2024), we introduce a feature-level distillation loss $\mathcal{L}_{\text{SG}}$, to ensure that the semantic representation of the student model ϵ_θ^S closely aligns with that of the teacher model ϵ_θ^T, at an intermediate feature level, formulated as:

$$\mathcal{L}_{\text{SG}} = \mathbb{E}\left[\sum_l ||M \odot F^l_{\epsilon_\theta^T} - M \odot \Phi(F^l_{\epsilon_\theta^S})||_2^2\right], \tag{3}$$

where $F^l_{\epsilon_\theta^T} := f^l_{\epsilon_\theta^T}(z_I, f_\mathcal{T})$ denotes the intermediate feature of ϵ_θ^T at layer l (and similarly, $F^l_{\epsilon_\theta^S}$ for the student model). Here, M is a binary mask with foreground and background regions set to 1 and other regions set to 0. M ensures that semantic supervision dose not interfere with the exploration of transparency details. The symbol $\odot$ represents the Hadamard product. The projection module Φ aligns the intermediate features of the two models, accounting for differences in their feature spaces due to distinct learning objectives. It consists of 3 learnable convolutional layers (Cv) and an intermediate LeakyReLU layer, defined as:

$$\Phi(F) = \text{Cv}_{3\times3}(\text{Cv}_{3\times3}(\text{LeakyReLU}(\text{Cv}_{1\times1}(F)))). \tag{4}$$

Transparency Mining Loss. To further encourage ϵ_θ^S to explore transparency details, we aim to extract and mine potential transparency information from ϵ_θ^T. We begin with the self-attention maps of ϵ_θ^T, a critical component for modeling intra-object consistency. Since ϵ_θ^T possesses well-learned semantic knowledge of alpha, the implicit transparency information used to identify such regions can serve as transparency cues, guiding ϵ_θ^S to further refine transparency details. Given the self-attention map $\mathbf{A}_f := \text{SoftMax}(\frac{\mathbf{Q}_f \cdot \mathbf{K}_f}{\sqrt{d}})$ derived from feature f, we introduce a self-attention-based transparency mining loss $\mathcal{L}_{\text{STM}}$, to constrain affinities specifically related to the alpha matte. To accomplish this, we propagate the GT alpha, α_{GT}, using the averaged self-attention map, $\mathbf{A}_{SA}$, computed across all self-attention layers at the same resolution. This propagation is performed via matrix multiplication ($\mathbf{A}_{SA} \otimes \alpha_{GT}$). We then minimize the difference between the propagated results of the teacher and student models. The rationale is as follows: $\mathbf{A}_{SA}$ acts as a transition matrix, while α_{GT} represents the current state. Their matrix multiplication transforms the current state into a resulting state. By constraining this resulting state, we indirectly regulate the α-related affinities within the self-attention map. This approach enhances the exploration of transparency details without incurring significant computational overhead, as it avoids constraining the entire self-attention map. We formulate $\mathcal{L}_{\text{STM}}$ as follows:

$$\mathcal{L}_{\text{STM}} = ||\mathbf{A}^T_{SA} \otimes \alpha_{GT} - \mathbf{A}^S_{SA} \otimes \alpha_{GT}||_2^2, \tag{5}$$

where $\otimes$ denotes matrix multiplication (not element-wise multiplication). Note that α_{GT} is down-sampled to the corresponding resolution using bilinear interpolation to satisfy the requirements for $\otimes$. We also extract potential transparency cues from the text expression using a similar cross-attention map distillation loss $\mathcal{L}_{\text{CTM}}$, which directly constrains the cross-attention maps between ϵ_θ^S and ϵ_θ^T, formulated as:

$$\mathcal{L}_{\text{CTM}} = ||\mathbf{A}^T_{CA} - \mathbf{A}^S_{CA}||_2^2. \tag{6}$$

For implementation, all attention maps are grouped by resolution and averaged to compute the loss. The final attention distillation loss is then calculated as the mean of the losses across all resolutions.

Total Objectives. In addition to the two key losses described earlier, we incorporate an L1 loss to train α^{lr} and a binary cross-entropy (BCE) loss to train R. These are defined as follows:

$$\mathcal{L}_{\alpha^{lr}} = ||\alpha^{lr} - \alpha^{lr}_{GT}||_1, \tag{7}$$

$$\mathcal{L}_R = -\widehat{M}\log(\frac{1}{1+e^R}) - (1-\widehat{M})\log(\frac{e^R}{1+e^R}), \tag{8}$$

where α^{lr}_{GT} represents the down-sampled GT alpha matte, and $\widehat{M} := 1 - M$ is a binary mask that indicates transparent regions. The final objective, $\mathcal{L}_{\epsilon_\theta^S}$, used to train ϵ_θ^S, is formulated as:

$$\mathcal{L}_{\epsilon_\theta^S} = \lambda_{\text{STM}}\mathcal{L}_{\text{STM}} + \lambda_{\text{CTM}}\mathcal{L}_{\text{CTM}} + \lambda_{\text{SG}}\mathcal{L}_{\text{SG}} + \lambda_{\alpha^{lr}}\mathcal{L}_{\alpha^{lr}} + \lambda_R\mathcal{L}_R, \tag{9}$$

where the λ coefficients are hyperparameters that balance the contribution of each loss term.

3.3 Structural Optimization on ϵ_θ^S

In our distillation framework, $\mathcal{L}_{\text{SG}}$ provides strong supervision signals to preserve localization knowledge by directly minimizing discrepancies in the semantic content of internal features. This highlights a key advantage of our approach: achieving a balance between maintaining performance and enhancing computational efficiency (see Sec. 3.2.2). To fully exploit this advantage, we apply two primary structural optimization techniques to the vanilla ϵ_θ^S (which initially shares the same architecture and parameters as ϵ_θ^T): block pruning and self-attention optimization.

Block Pruning. We prune redundant blocks from the vanilla student model following Kim et al. (2024), resulting in a model with fewer parameters. Guided by $\mathcal{L}_{\text{SG}}$, the performance degradation caused by this pruning is effectively mitigated. Additionally, the other loss terms encourage the student model to further assimilate soft grounding knowledge.

Self-Attention Optimization. The self-attention mechanism in Transformer blocks is a critical component for modeling long-range dependencies. However, the matrix multiplication operations in self-attention are computationally expensive, resulting in an overall time complexity of $O(n^2d + nd^2)$, which becomes particularly burdensome at higher resolutions. Nevertheless, the soft grounding task has more relaxed accuracy requirements, as the learned outputs (α^{lr} and R) provide sufficient flexibility to tolerate errors in the subsequent alpha enhancement process. Inspired by this, we propose that the self-attention operation can be optimized into a more computationally efficient form by learning asymmetric sparse correspondences.

Specifically, we argue that the dense affinity matrix derived from the self-attention map $\mathbf{A}_f$ can be simplified into a unidirectional sparse representation. To achieve this, we can learn a smaller set of representative feature tokens and use their affinities to approximate the affinities between the original feature tokens and others within a relatively small region. Thus, we introduce a learnable down-sampling operation ϕ, which is applied to f when calculating $\mathbf{K}$ and $\mathbf{V}$ (*i.e.*, $\mathbf{K}^S = W_K\phi(f), \mathbf{V}^S = W_V\phi(f)$), while $\mathbf{Q}$ remains unchanged. ϕ can be implemented using a single convolution layer with a kernel size and stride of $k \times k$, where k is chosen from $\{2^i | 1 \leq i \leq \log_2 h_{\mathbf{A}_f^S}\}$, and $h_{\mathbf{A}_f^S}$ denotes the height of the square matrix $\mathbf{A}_f^S$, theoretically. The quantitative analysis of the reduction in computational cost can be found in the **Appendix**. Note that $\mathcal{L}_{\text{STM}}$ contributes to preserving intra-object cohesion and textural information within the optimized self-attention mechanism.

3.4 From Soft Grounding to Alpha Matte

To up-sample α^{lr} from a resolution of $\frac{H}{8} \times \frac{W}{8}$ to α at $H \times W$, we adopt the detail encoder-decoder structure proposed in Yao et al. (2024a). The encoder takes the concatenation of (I, α^{lr}, R) as input to extract detail features. Before that, α^{lr} and R are interpolated to match the resolution of I, and R is processed through a sigmoid function. The detail features from the encoder's final layer are concatenated with features extracted from ϵ_θ^S and then passed to a decoder to predict the final alpha matte α. For more efficient training, we freeze ϵ_θ^S and train only the parameters of the encoder and decoder. Additionally, a single learnable ConvBNReLU block is applied to the output features from ϵ_θ^S before concatenation to align the feature spaces. This process is supervised using an L1 loss and a Laplacian loss following Li et al. (2022).

Extend to High-resolution Inference. We implement a simple upsampling module based on sparse convolution (Contributors, 2022), integrated with the matting decoder, to enable inference at arbitrary high resolutions (up to 2K). This sparse convolution approach effectively reduces computational cost and memory usage during the upsampling process.

4 Experiment

4.1 Implementation Details

Data Acquisition. The data used to train our model comprises 4 matting datasets (RefMatte (Li et al., 2023a), P3M10K (Li et al., 2021a), AM2K (Li et al., 2022), RM1K (Wang et al., 2023b)), and 1 grounding segmentation dataset (RefCOCO (Kazemzadeh et al., 2014)). Considering there are no text annotations for P3M10K, AM2K, and RM1K, we adopt BLIP2 (Li et al., 2023b) to generate text annotations for each sample in these datasets by guiding the BLIP2 model to describe the appearance of the object in the image. For samples from RefMatte and RefCOCO during training, we randomly select one expression for each object if multiple expression annotations exist for the same object. Since there are no matting-level annotations for RefCOCO, we first generate the pseudo trimap according to the mask annotations using morphological operations, then use Yao et al. (2024a) to obtain alpha annotations.

Training Details. All stages of our model's training process adopt a consistent data scheduling strategy. Specifically, we train the model on RefMatte during odd-numbered iterations and on RefCOCO in every even iteration. We also insert a special iteration after every 4 iterations to perform training on P3M10K, AM2K, and RM1K. We randomly select 1 dataset among the 3 to train our model in this special iteration. We set the kernel size of the morphological operation to 15, and we set $(\lambda_{\text{STM}}, \lambda_{\text{CTM}}, \lambda_{\text{SG}}, \lambda_{\alpha^{lr}}, \lambda_R)$ to $(10, 0.1, 0.5, 10, 1)$. Other training settings, including batch size, learning rate, total iterations, and rationales behind setting λs, can be found in the **Appendix**.

Details about Optimized ϵ_θ^S. To thoroughly evaluate our proposed framework, we adopt the most extreme pruning scheme proposed in Kim et al. (2024) to build ϵ_θ^S, *i.e.*the *tiny* setting, which removes 5 blocks in the encoder and 5 blocks in the decoder, and removes the entire middle block in ϵ_θ. For optimizing self-attention (SA), we directly remove all SA layer at resolution of 64×64 following Zhao et al. (2023b), and set the k value for SA layers at 32×32 and 16×16 (k_{32^2}, k_{16^2}) as $k_{32^2} = k_{16^2} = 2$.

4.2 Evaluate Metrics

Following Rhemann et al. (2009), we adopt 4 matting metrics for evaluation, including SAD, MSE, Gradient Error (GRAD), and Connectivity Error (CONN). These metrics are scaled by 10^3, 10^{-3}, 10^3, 10^3, respectively. Lower values indicate better performance across all metrics.

4.3 Comparison on Soft Grounding

Baselines. To comprehensively evaluate our model's performance, we compare it with three types of baselines.

- **Text-guided matting methods.** We select CLIPMat (Li et al., 2023a) as a fundamental baseline, which solves the soft grounding problem directly and shares similar settings with ours. Since CLIPMat is currently closed-source, we re-implement it and use the same training settings as ours.
- **Visual grounding with interactive matting methods.** We select the representative GroundingDINO (GDINO) (Liu et al., 2024b) as the visual grounding method, which produces the bounding box of the foreground object according to the text input. Then we select 3 recent interactive matting methods to obtain the alpha given the bounding box, including MatAny (Yao et al., 2024b), MAM (Li et al., 2024b), and SmartMat (Ye et al., 2024).
- **Grounding segmentation with mask-guided matting methods.** We also select the latest grounding segmentation method PSALM (Zhang et al., 2024) and the SD-based referring segmentation method RefVPD (Zhao et al., 2023a) to generate a mask for the foreground object based on text input. We then apply two mask-guided matting methods to derive the alpha matte from the mask, including MaGGIe (Huynh et al., 2024) and MGMat (Yu et al., 2021).

For fairness, all baselines with cascaded structure, including the visual grounding part (except PSALM and GDINO) and matting part, are aligned to our training data through fine-tuning. Note that PSALM and GDINO are trained on very large-scale datasets. We attempted to fine-tune them directly on our relatively small-scale training dataset, but this yielded poorer results. Consequently, we use their original weights for further comparison.

Benchmarks. We apply two referring natural matting benchmarks, including RefMatte-Test (Li et al., 2023a) and RefMatte-RW100 (Li et al., 2023a), for soft grounding evaluation. Here, the former is a composition dataset (6,243 instances among 2,500 images) and the latter is a real-world dataset (221 instances among 100 images). Every instance in these two benchmarks has 4 different expressions, so we evaluate all baselines and ours using all expressions and report the average result among these 4 expressions. During evaluation, the input resolution for all methods is set to 512×512, and the metrics are also calculated on this

Table 1: **Comparison on soft grounding task.** Our model outperforms all baselines on two referring natural image matting benchmarks (RefMatte-Testset and RefMatte-RW100) with faster inference time, demonstrating the effectiveness and practicality of our method.

Methods		RefMatte-Testset				RefMatte-RW100				Inference
		SAD↓	MSE↓	GRAD↓	CONN↓	SAD↓	MSE↓	GRAD↓	CONN↓	Time↓ (ms)
Grounding Model + Interactive Matting	GDINO+MatAny	15.52	0.0561	8.41	3.61	19.92	0.0727	9.57	7.71	984.57
	GDINO+MAM	15.16	0.0552	8.92	4.59	16.96	0.0626	8.56	10.78	496.30
	GDINO+SmartMat	11.69	0.0403	7.91	1.78	16.91	0.0616	9.30	5.54	122.29
Gounding Segmentation Model + Mask-guided Matting	PSALM+MaGGIe	8.71	0.0299	7.63	2.67	9.90	0.0349	8.24	**3.75**	287.79
	PSALM+MGMat	8.55	0.0301	7.02	2.57	9.41	0.034	6.83	3.81	272.21
	RefVPD+MaGGIe	9.01	0.0308	8.36	3.23	11.41	0.0399	9.64	5.63	220.79
	RefVPD+MGMat	8.98	0.0315	7.89	3.33	10.32	0.0374	7.66	5.37	205.21
Soft Grounding	CLIPMat	26.56	0.1181	14.84	8.15	35.12	0.1951	21.01	18.97	102.74
	Ours	**3.19**	**0.0098**	**3.55**	**1.69**	**7.37**	**0.0264**	**6.55**	5.31	**95.14**

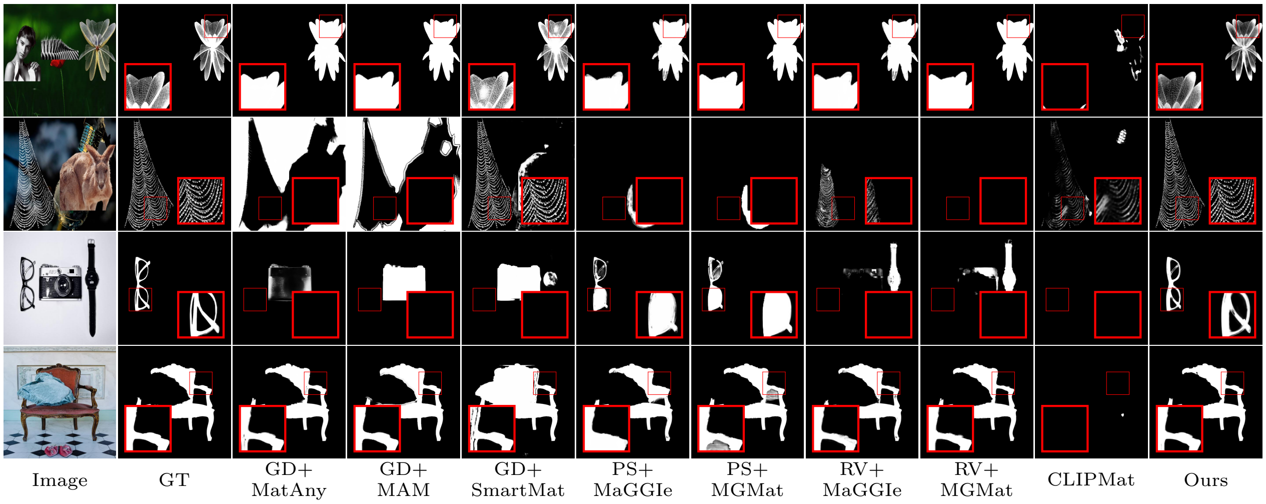


Figure 4: **Qualitative comparison on soft grounding task.** The text inputs from top to bottom are: 1) *the insect which is darkgray*; 2) *the white and non-salient and transparent net*; 3) *a modern-designed glass with a black frame*; 4) *the red chair made of wood.*

resolution. We also report the average inference time per sample in milliseconds, using the same machine with a single RTX 3090.

Quantitative Results. We show the quantitative comparison on soft grounding in Tab. 1. We found a significant performance gap between CLIPMat and ours since the feature and cross-modal prior within SD model make it easier to solve the soft grounding problem compared with CLIP. Our model also achieves the best performance on most of the metrics compared with the cascaded baselines, even when training data is aligned among all matting networks. Although adopting PSALM, which is based on Large Multi-modal Model (LMM) for grounding segmentation, can obtain lower connectivity error on RW100 benchmark, the other three metrics are still worse than ours and have relatively high inference time. Instead, our framework can teach a structure-optimized model to achieve the best performance with lower inference time, showing the effectiveness of our proposed framework.

Qualitative Results. Some qualitative comparisons are shown in Fig. 10. Without properly modeling soft grounding, cascading-based baselines often perform poorly with incorrect semantic and low-quality details as shown in the first two rows. Although some baselines adopt SAM (Kirillov et al., 2023) or LMM (Liu et al., 2024a), they still show sub-optimal performance on soft grounding in the real-world (last two rows). More comparison results can be found in the **Appendix**.

4.4 Comparison on Generalization Ability

Introducing text prior can unify various matting tasks into soft grounding. To evaluate the generalization ability of our model on such fine-grained matting tasks, we select several specialist methods in their own task

Table 2: **Quantitative comparison on generalization ability across different matting tasks.** Top-2 results are marked in **bold** and underlined. Task-specific methods only perform well on their own task, while ours can generalize to various matting tasks using text-based guidance **without fine-tuning**.

Method	Type	AM2K (animal)		P3M-P (portrait)		RefMatte-Test (referring natural)	
		SAD↓	MSE↓	SAD↓	MSE↓	SAD↓	MSE↓
GFM (Li et al., 2022)	animal	**12.08**	**0.0035**	347.44	0.2001	239.33	0.1335
P3M-ViTAE (Ma et al., 2023)	portrait	40.43	0.0204	**6.59**	**0.0015**	290.05	0.1616
GenPercept (Xu et al., 2024a)	portrait	19.04	0.0049	11.02	0.0025	269.50	0.1456
AIM (Li et al., 2021b)	natural	28.25	0.0101	45.41	0.0207	336.76	0.1840
Ours		13.81	0.0045	9.53	0.0030	**24.19**	**0.0112**

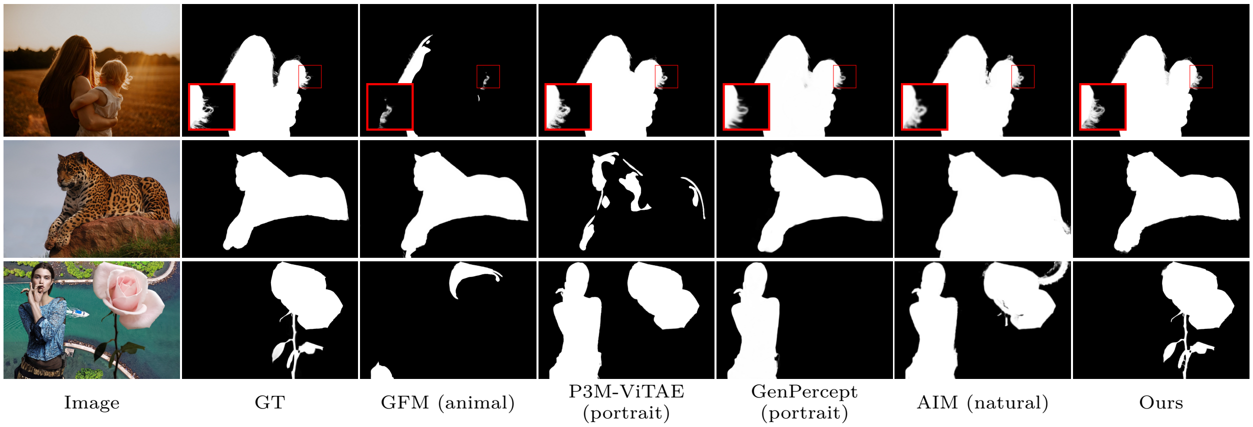


Figure 5: **Qualitative comparison on generalization ability across different matting tasks.** The text inputs used in ours are, from top to bottom: 1) *mother and daughter in field at sunset*; 2) *a jaguar is sitting on top of a rock*; 3) *the works which are thistle and non-transparent*.

to compare with our model quantitatively (Tab. 2). All baselines here are directly applied using the officially released weight, and our model is tested without task-specific tuning. All the metrics are calculated in full resolution. We found that although the baselines excel in their designated tasks, they often lack generalization and struggle with referring matting across diverse categories. In contrast, our model, leveraging a text prior, generalizes effectively to various matting tasks, though it slightly lags behind task-specific experts in their respective domains. Most importantly, our model shows unrivaled performance on referring natural matting tasks, where other baselines falter. We also showcase some qualitative results in Fig. 5. We found GFM and P3M lack generalization, while GenPercept and AIM predict over-smoothed alpha. All of them have poor performance on the soft grounding task.

4.5 Ablation Studies

To fully demonstrate the necessity of our key design and modules in our model, we conduct following ablation studies via different training settings and quantify results in Tab. 3, using RefMatte-RW100 as the benchmark.

Problem Disentanglement. First, directly training SD for soft grounding proves highly challenging, resulting in suboptimal performance (row 1) compared to a disentangled formulation.

Knowledge Distillation (KD). Next, we explore a two-stage approach where SD is pre-trained for localization and then fine-tuned for soft grounding (row 2). This significantly improves performance but remains inferior to our complete KD framework (row 5). These results highlight the importance of preserving semantic knowledge and explicitly separating the two sub-objectives. Furthermore, fine-tuning a lightweight ϵ_θ^S without KD leads to a substantial performance drop (row 3), underscoring the critical role of KD in maintaining a balance between efficiency and performance.

Table 3: **Ablation studies.** For instance, (ϵ_θ^T🔥, SG) means training ϵ_θ^T on soft grounding task. "ϵ_θ^T❄-ϵ_θ^S🔥" denotes the distillation framework. ϵ_θ^S is structural optimized to lightweight in row 2˜6.

#	Training settings (SG=Soft Grounding, Local.=Localization)	Φ	Mat. E-D	SAD↓	MSE↓	GRAD↓	CONN↓
1	(ϵ_θ^T🔥, SG)			17.50	0.0645	9.45	8.27
2	(ϵ_θ^T🔥, Local.), then fine-tune (ϵ_θ^T🔥, SG)			9.75	0.0375	7.89	6.48
3	(ϵ_θ^T🔥, Local.), then fine-tune (ϵ_θ^S🔥, SG)			13.27	0.0487	9.01	6.73
4	(ϵ_θ^T🔥, Local.), (ϵ_θ^T❄-ϵ_θ^S🔥, SG)			15.96	0.0534	9.26	7.55
5	(ϵ_θ^T🔥, Local.), (ϵ_θ^T❄-ϵ_θ^S🔥, SG)	✓		7.63	0.0267	6.98	5.35
6	(ϵ_θ^T🔥, Local.), (ϵ_θ^T❄-ϵ_θ^S🔥, SG)	✓	✓	7.37	0.0264	6.55	5.31
7	(ϵ_θ^T🔥, Local.), (ϵ_θ^T❄-ϵ_θ^S(w/o. opti.)🔥, SG)	✓	✓	6.30	0.0223	5.52	4.33

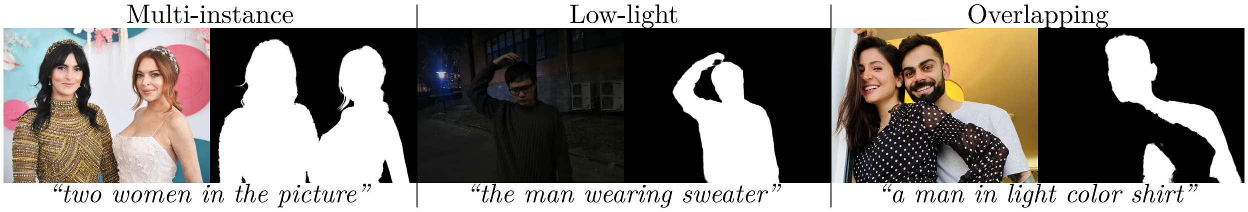

Multi-instance
Low-light
Overlapping
"two women in the picture"
"the man wearing sweater"
"a man in light color shirt"

Figure 6: **Robustness Evaluation.** Our method demonstrates strong performance across a variety of challenging scenarios.

Other Network Components. We also observe that removing the feature projection module (Φ) confuses ϵ_θ^S during distillation, as ϵ_θ^T and ϵ_θ^S have distinct learning objectives, leading to degraded performance (row 4). Additionally, integrating a dedicated matting encoder and decoder (Mat. E-D) (row 6, complete framework) further improves the quality of the predicted alpha matte.

Structural Optimization. Lastly, we evaluate the impact of structural optimization. While the vanilla ϵ_θ^S without structural optimization (row 7) achieves higher performance, its inference time is significantly longer (143ms per image under the same setting in Sec. 4.3). This further demonstrates that our framework enables lightweight models to achieve strong performance with improved efficiency.

More ablation studies on hyperparameters, objectives, and SD versions are provided in the **Appendix**.

4.6 Robustness Evaluation and Limitations

Given accurate and well-structured text prompts, our model demonstrates strong robustness against various typical disturbances (see Fig. 6). However, like all prompt-based systems, it is susceptible to vague or ambiguous prompts, which can degrade performance (see Fig. 7). Future work could explore adaptive guidance strategies to enhance robustness and flexibility across a broader range of matting scenarios.

5 Conclusion

In this paper, we unified various matting tasks as a soft grounding problem, addressing both instance-level localization and transparency estimation using the visual-language capabilities of Stable Diffusion. Instead of directly training SD, we introduced a distillation framework that separates localization and transparency tasks, improving performance in complex scenarios with multiple objects or intricate transparency patterns. This framework also enables a more efficient student model, balancing complexity and accuracy. Extensive evaluations demonstrate our method's effectiveness, with competitive speed and strong generalization across diverse matting categories.

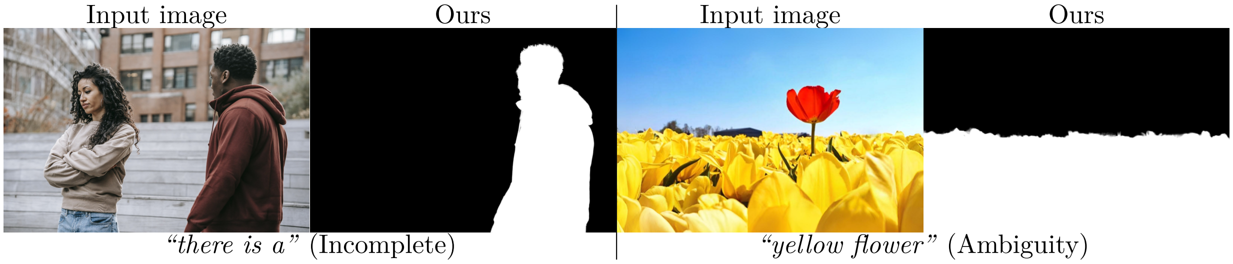


Figure 7: **Some failure cases of our method caused by ambiguous prompts.**

References

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pp. 22563–22575, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pp. 18392–18402, 2023.

Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. In *CVPRW*, 2023.

Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. *IEEE TPAMI*, 35(9):2175–2188, 2013.

Spconv Contributors. Spconv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, 132(2):581–595, 2024.

Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, pp. 5958–5966, 2018.

He Guo, Zixuan Ye, Zhiguo Cao, and Hao Lu. In-context matting. In *CVPR*, pp. 3711–3720, 2024.

Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. In *CVPR*, pp. 2049–2056, 2011.

Qiqi Hou and Feng Liu. Context-aware image matting for simultaneous foreground and alpha estimation. In *ICCV*, pp. 4130–4139, 2019.

Yihan Hu, Yiheng Lin, Wei Wang, Yao Zhao, Yunchao Wei, and Humphrey Shi. Diffusion for natural image matting. In *ECCV*, pp. 181–199, 2024.

Chuong Huynh, Seoung Wug Oh, Abhinav Shrivastava, and Joon-Young Lee. Maggie: Masked guided gradual human instance matting. In *CVPR*, pp. 3870–3879, 2024.

Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *ICCV*, pp. 21741–21752, 2023.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pp. 787–798, 2014.

Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *ECCV*, pp. 381–399, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pp. 4015–4026, 2023.

Hsin-Ying Lee, Hung-Yu Tseng, and Ming-Hsuan Yang. Exploiting diffusion prior for generalizable dense prediction. In *CVPR*, pp. 7861–7871, 2024.

Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE TPAMI*, 30(2):228–242, 2007.

Jiachen Li, Roberto Henschel, Vidit Goel, Marianna Ohanyan, Shant Navasardyan, and Humphrey Shi. Video instance matting. In *WACV*, pp. 6668–6677, 2024a.

Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. In *CVPRW*, pp. 1775–1785, 2024b.

Jizhizi Li, Sihan Ma, Jing Zhang, and Dacheng Tao. Privacy-preserving portrait matting. In *ACM MM*, pp. 3501–3509, 2021a.

Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. *IJCAI*, pp. 800–806, 2021b.

Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Bridging composite and real: towards end-to-end deep image matting. *IJCV*, 130:246–266, 2022.

Jizhizi Li, Jing Zhang, and Dacheng Tao. Referring image matting. In *CVPR*, pp. 22448–22457, 2023a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742, 2023b.

Xiaodi Li, Zongxin Yang, Ruijie Quan, and Yi Yang. DRIP: Unleashing diffusion priors for joint foreground and alpha prediction in image matting. In *NeurIPS*, 2024c.

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, pp. 8762–8771, 2021.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *CVPR*, pp. 10034–10043, 2020.

Sihan Ma, Jizhizi Li, Jing Zhang, He Zhang, and Dacheng Tao. Rethinking portrait matting with privacy preserving. *IJCV*, 131(8):2172–2197, 2023.

GyuTae Park, SungJoon Son, JaeYoung Yoo, SeHo Kim, and Nojun Kwak. Matteformer: Transformer-based image matting via prior-tokens. In *CVPR*, pp. 11696–11706, 2022.

Kwanyong Park, Sanghyun Woo, Seoung Wug Oh, In So Kweon, and Joon-Young Lee. Mask-guided matting in the wild. In *CVPR*, pp. 1992–2001, 2023.

Thomas Porter and Tom Duff. Compositing digital images. In *SIGGRAPH*, pp. 253–259, 1984.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. Make-a-story: Visual memory conditioned consistent story generation. In *CVPR*, pp. 2493–2502, 2023.

Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *CVPR*, pp. 1826–1833, 2009.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pp. 22500–22510, 2023.

Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *CVPR*, pp. 2291–2300, 2020.

Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *CVPR*, pp. 8839–8849, 2024.

Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Human instance matting via mutual guidance and multi-instance refinement. In *CVPR*, pp. 2647–2656, 2022.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pp. 12619–12629, 2023a.

Yanfeng Wang, Lv Tang, Yijie Zhong, and Bo Li. From composited to real-world: Transformer-based natural image matting. *IEEE TCSVT*, 2023b.

Zhixiang Wang, Baiang Li, Jian Wang, Yu-Lun Liu, Jinwei Gu, Yung-Yu Chuang, and Shin'ichi Satoh. Matting by generation. In *SIGGRAPH*, pp. 1–11, 2024.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Hanqing Zhao, Weiming Zhang, and Nenghai Yu. Improved image matting via real-time user clicks and uncertainty estimation. In *CVPR*, pp. 15374–15383, 2021.

Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, pp. 4974–4984, 2022.

Ruihao Xia, Yu Liang, Peng-Tao Jiang, Hao Zhang, Qianru Sun, Yang Tang, Bo Li, and Pan Zhou. Towards natural image matting in the wild via real-scenario prior. *arXiv preprint arXiv:2410.06593*, 2024.

Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. Diffusion models trained with large data are transferable visual models. *arXiv preprint arXiv:2403.06090*, 2024a.

Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pp. 2955–2966, 2023a.

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, pp. 2970–2979, 2017.

Yangyang Xu, Shengfeng He, Wenqi Shao, Kwan-Yee K Wong, Yu Qiao, and Ping Luo. Diffusionmat: Alpha matting as sequential refinement learning. *arXiv preprint arXiv:2311.13535*, 2023b.

Yong Xu, Xin Yao, Baoling Liu, Yuhui Quan, and Hui Ji. Text-guided portrait image matting. *IEEE Transactions on Artificial Intelligence*, 2024b.

Peiqing Yang, Shangchen Zhou, Jixin Zhao, Qingyi Tao, and Chen Change Loy. MatAnyone: Stable video matting with consistent memory propagation. In *arXiv preprint arXiv:2501.14677*, 2025.

Stephen DH Yang, Bin Wang, Weijia Li, YiQi Lin, and Conghui He. Unified interactive image matting. *arXiv preprint arXiv:2205.08324*, 2022.

Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pre-trained plain vision transformers. *Information Fusion*, 103:102091, 2024a.

Jingfeng Yao, Xinggang Wang, Lang Ye, and Wenyu Liu. Matte anything: Interactive natural image matting with segment anything model. *Image and Vision Computing*, 147:105067, 2024b.

Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *CVPR*, pp. 10502–10511, 2019.

Zixuan Ye, Wenze Liu, He Guo, Yujia Liang, Chaoyi Hong, Hao Lu, and Zhiguo Cao. Unifying automatic and interactive matting with pretrained vits. In *CVPR*, pp. 25585–25594, 2024.

Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. Mask guided matting via progressive refinement network. In *CVPR*, pp. 1154–1163, 2021.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *ECCV*, pp. 74–91, 2024.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, pp. 5729–5739, 2023a.

Yang Zhao, Yanwu Xu, Zhisheng Xiao, and Tingbo Hou. Mobilediffusion: Subsecond text-to-image generation on mobile devices. *arXiv preprint arXiv:2311.16567*, 2023b.

6 Appendix

6.1 Analysis of Self-Attention Optimization

Here, we compare the computational overhead of vanilla self-attention (SA) and the proposed sparse self-attention (Sp.SA) within the framework of our method. **Complexity of Vanilla SA.** Let n represent the number of input tokens and d the dimension of the token embedding. The computational complexity of a single SA operation—*i.e.*, the total number of addition and multiplication operations ($\mathcal{O}_{\text{SA}}$)—can be calculated as follows:

$$\begin{aligned}\mathcal{O}_{\text{SA}} &= \underbrace{2nd^2 \times 3}_{Q,K,V} + \underbrace{2n^2d}_{Q \cdot K} + \underbrace{n^2}_{(\sqrt{d})^{-1}} + \underbrace{3n^2-1}_{\text{Softmax}} + \underbrace{2n^2d}_{AV} \\ &= 6nd^2 + 4n^2d + 4n^2 - 1\end{aligned} \tag{10}$$

For instance, given the input feature f, computing Q involves multiplying $f \in \mathbb{R}^{n \times d}$ and $W_Q \in \mathbb{R}^{d \times d}$, resulting in $n \times d \times d$ addition operations and $n \times d \times d$ multiplication operations. Hence, the total number for calculating Q is $2nd^2$. The same applies to the computations of $Q \cdot K$ and AV. Additionally, since the normalization $(\times(\sqrt{d})^{-1})$ and Softmax operations act on an $n \times n$ matrix, their complexities are n^2 and $\underbrace{n^2}_{\text{exponential}} + \underbrace{(n^2-1)}_{\text{addition}} + \underbrace{n^2}_{\text{division}} = 3n^2 - 1$, respectively[*].

Complexity of Optimized SA. As mentioned in Sec. 3.3 in the main paper, introducing the down-sampled convolution ϕ with kernel size and stride both equal to $k \times k$ results in K^S and V^S with a sequence length k^2 times shorter than the original K and V. Let j ($j = k^2$) be the scale factor of this operation; the corresponding complexity $\mathcal{O}_{\text{Sp.SA}}$ is given as:

[*] FLOPs for exponential operation can be seen as 1 due to common instruction optimization and hardware acceleration rather than using repeating multiplication.

$$\begin{aligned}\mathcal{O}_{\text{Sp.SA}} &= \underbrace{2nd^2}_{Q} + \underbrace{2 \times \frac{n}{j} \times d^2 \times 2}_{K^S, V^S \text{ with } \phi \text{ applied}} + \underbrace{2nd \times \frac{n}{j}}_{Q \cdot K^S} + \underbrace{n \times \frac{n}{j}}_{(\sqrt{d})^{-1}} \\ &+ \underbrace{3n \times \frac{n}{j} - 1}_{\text{Softmax}} + \underbrace{2n \times \frac{n}{j} \times d}_{AV^S} + \underbrace{j \times \frac{n}{j} \times d \times d}_{\text{Convolution operation } \phi} \\ &= (3 + \frac{4}{j})nd^2 + \frac{4n^2d}{j} + \frac{4n^2}{j} - 1\end{aligned} \tag{11}$$

In this work, we set $k_{32^2} = k_{16^2} = 2$ for SA layers at resolutions of 32×32 and 16×16 respectively, with number of channels $d_{32^2} = 640, d_{16^2} = 1280$. Then, the fraction of complexity reduction is calculated as:

$$\begin{aligned}\frac{\mathcal{O}_{\text{Sp.SA}}(n = 32^2, d = 640, j = 2^2)}{\mathcal{O}_{\text{SA}}(n = 32^2, d = 640)} &\approx 0.4515 \\ \frac{\mathcal{O}_{\text{Sp.SA}}(n = 16^2, d = 1280, j = 2^2)}{\mathcal{O}_{\text{SA}}(n = 16^2, d = 1280)} &\approx 0.6176\end{aligned} \tag{12}$$

The results show that the proposed Sp.SA can reduce the computational complexity by 54.85% and 38.24% for the SA layers at 32×32 and 16×16 resolution, respectively. Given the same j, the complexity reduction is more significant for the SA layer with a longer sequence length, which aligns with the intuition that the computational overhead of SA is proportional to the square of the sequence length.

6.2 More Details on Training

Data Pre-processing. All the images are resized to 512×512 for training in all stages. For composition samples from RefMatte (Li et al., 2023a), we use the same augmentation strategy as in Li et al. (2022) to reduce the discrepancy between synthetic and real data.

Rationals of Hyperparamter Setting. For the kernel size of the morphological operation K_{mor}, we set it to 15 to balance overestimation (if too large) and underestimation (if too small) of the transparent region, aligning with common practice in matting methods. For the cross-attention loss $\mathcal{L}_{\text{CTM}}$, we use `sum` reduction to address the presence of many zero matrices in the cross-attention maps (e.g., padding tokens), leading to a smaller λ_{CTM} of 0.1. During training, $\mathcal{L}_{\text{STM}}$ and $\mathcal{L}_{\alpha^{lr}}$ are relatively small, while $\mathcal{L}_{\text{SG}}$ is larger under `mean` reduction. To balance these, we set $(\lambda_{\text{STM}}, \lambda_{\alpha^{lr}}, \lambda_{\text{SG}})$ to $(10, 10, 0.5)$.

Training Hyperparameters. We show the training parameters in Tab. 4. We adopt AdamW (Loshchilov & Hutter, 2019) as the optimizer with weight decay as 0.01. All the training work is done on NVIDIA A100 80GB GPU(s). For the scheduler, we adopt a LambdaLR scheduler with several specified milestones. When the milestone is not reached, we use the lambda function $f(x) = (1 - \frac{x}{\text{total iterations}})^{0.9}$ to adjust the learning rate. Otherwise, the learning rate is directly adjusted to the preseted value.

Table 4: Hyperparameters for all training stages.

Training Stage	Initial Learning Rate	Total Iterations	Total Batch Size	Scheduler Value	Scheduler Milestone	GPUs (A100)	Learnable Parameters	Training Time
Soft Grounding in Semantic	5e-5	50000	32	[0.5, 0.25]	[15000, 40000]	4	861M	18.5h
Soft Grounding in Transparency	5e-5	50000	32	[0.4, 0.25]	[10000, 35000]	2	509M	20.7h
From Soft Grounding to Alpha Matte	4e-4	50000	16	[0.1, 0.05]	[10000, 35000]	1	2.67M	12.7h

6.3 More about Sparse Up-sampling Module

The design of the sparse up-sampling module is inspired by the progressive refinement decoder proposed in Yu et al. (2021), consisting of two groups of interpolation-convolution groups. All the vanilla convolutional layers are replaced by sparse convolutional layers (Contributors, 2022) in the up-sampling module. The input of each group is the concatenation of the up-interpolated alpha matte and the context feature from the matting decoder (all with a resolution of 512×512), along with the original image at the target resolution. The sparse convolution layer performs calculations only on areas where the alpha matte value is greater than 0.01 and less than 0.99. To train the module, we attach it to the pre-trained matting decoder and train the entire network using the same loss function as the matting decoder. The learning rate of the module is set to $2e-4$, and the total number of iterations is set to $10,000$ with a batch size of 16.

6.4 More Ablation Studies

Loss Functions. We also explore the effect of each loss adopted in the distillation process in Tab. 5 using RefMatte-RW100 as the benchmark. Since $\mathcal{L}_{\text{SG}}$ is the key loss for semantic guidance from $\epsilon^T\theta$, omitting it will lead to a significant performance drop. $\mathcal{L}_{\text{STM}}$ and $\mathcal{L}_{\text{CTM}}$ also evidently improve the final performance, demonstrating the effectiveness of transparency exploration and retention of attention information for the optimized ϵ_θ^S. Furthermore, the learned transparent confidence map R via $\mathcal{L}_{\text{R}}$ also facilitates the matting E-D in predicting more accurate alpha mattes.

Table 5: **Ablation study on different losses.**

$\mathcal{L}_{\alpha^{lr}}$	$\mathcal{L}_{\text{SG}}$	$\mathcal{L}_{\text{STM}}$	$\mathcal{L}_{\text{CTM}}$	$\mathcal{L}_{\text{R}}$	SAD↓	MSE↓	GRAD↓	CONN↓
✓					14.26	0.0465	8.95	7.09
✓	✓				8.40	0.0297	7.25	6.11
✓	✓	✓			8.13	0.0284	7.06	5.98
✓	✓	✓	✓		7.96	0.0279	6.91	5.80
✓	✓	✓	✓	✓	7.37	0.0264	6.55	5.31

Self-Attention Optimization Kernel. We further evaluate the impact of using different kernel sizes k for the down-sampled convolution ϕ in the optimization of SA at various resolutions. Our default setting removes SA layers at a resolution of 64×64 (denoted as $k_{64^2} = -1$) and sets $k_{32^2} = k_{16^2} = 2$. Therefore, we attempt to increase the kernel size at resolutions of 32×32 and 16×16, and also try restoring SA layers at a resolution of 64×64 with $k_{64^2} = 2$ instead of removing them entirely, resulting in three different settings.

Table 6: **Ablation study on kernel size k in Self-Attention Optimization.** Here -1 means remove the SA layers directly.

k_{64^2}	k_{32^2}	k_{16^2}	SAD↓	MSE↓	GRAD↓	CONN↓	Inference Time (ms)
-1	2	2	7.37	0.0264	6.55	5.31	95.14
-1	2	4	8.57	0.0301	7.24	6.05	94.90
-1	4	2	7.65	0.0272	6.70	5.54	94.69
2	2	2	7.13	0.0255	6.37	5.19	130.03

In Tab. 6, we report the quantitative results of these three settings (the last three rows) alongside our default setting (the first row), using RefMatte-RW100 as the benchmark. The inference time per sample is also evaluated, following the same setup as in Sec. 4.3 of the main paper. We found that using a larger kernel size at resolutions of 32×32 and 16×16 can lead to performance degradation, as the over-optimized SA layers lose their ability to capture global context and intra-object cohesion information. This effect is particularly noticeable at a resolution of 16×16, where increasing k causes a significant performance drop. Moreover, the running time is not substantially reduced in either case. We also found that restoring SA layers at a resolution of 64×64 with $k_{64^2} = 2$ yields only marginal improvements while significantly slowing down the model.

Sparse Up-sampling Module. We conduct the ablation study of the sparse up-sampling module on the RefMatte-Test dataset in full resolution, which consists of more natural image matting samples. The quantitative result is shown in Tab. 7. We found that three metrics (SAD, MSE, GRAD) are improved with the sparse up-sampling module, while the CONN metric is slightly higher. This is mainly because an over-smoothed alpha leads to a lower connectivity error value. We then show the qualitative comparison in Fig.8.

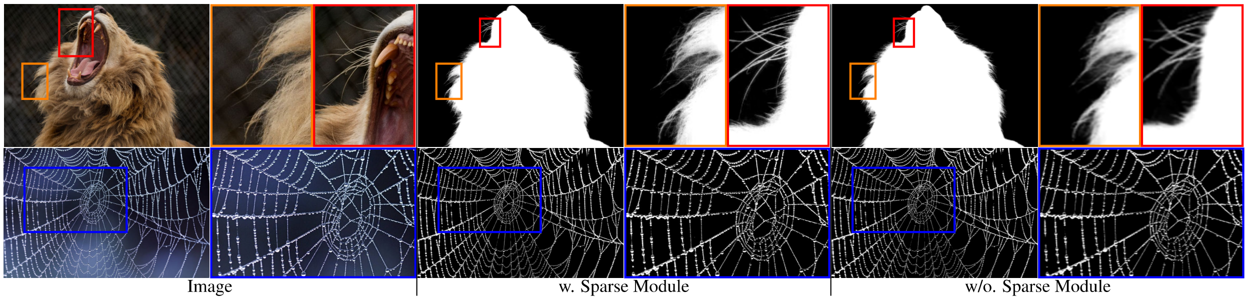


Figure 8: **Qualitative ablation comparison of the sparse up-sampling module (SUM).** The text inputs for the examples from left to right are: 1) *a lion yawning with its mouth open*; 2) *a spider web is shown.* SUM preserves more high-frequency textures, resulting in a clearer alpha matte.

Despite the higher connectivity error, SUM can produce clearer alpha with finer details, demonstrating the effectiveness of this module in extending our model to arbitrary high-resolution inference, with more accurate matting texture preserved compared with direct up-sampling.

Table 7: **Ablation study on sparse up-sample module (SUM).**

	SAD↓	MSE↓	GRAD↓	CONN↓
Ours (w/o. SUM)	24.87	0.0114	16.43	12.58
Ours (w. SUM)	24.19	0.0112	14.94	12.91

Hyperparameters. We analyze the impact of hyperparameters in the training process using RefMatte-RW100 as the benchmark (see Tab. 8). We evaluate three alternative settings and find that our default configuration (row 1) consistently outperforms them, demonstrating its effectiveness.

Table 8: **Ablation study on hyperparameters.**

λ_{STM}	λ_{CTM}	λ_{SG}	$\lambda_{\alpha^{lr}}$	λ_R	K_{mor}	SAD↓	MSE↓	GRAD↓	CONN↓
10	0.1	0.5	10	1	15	7.37	0.0264	6.55	5.31
10	0.1	0.5	10	1	5	9.32	0.0354	8.57	6.61
1	0.1	0.5	1	1	15	7.65	0.0271	6.63	5.32
10	0.1	1	10	1	15	7.41	0.0268	6.55	5.29

Stable Diffusion Versions. To further assess the generalizability of our framework, we apply our method to teach two additional versions of Stable Diffusion (SD v2.0 and SD v2.1) for soft grounding while keeping all other experimental settings unchanged. The evaluation results on RefMatte-RW100 (see Tab. 9) show that compared to our default setting (SD v1.5), SD v2.x achieves better performance, highlighting the broad applicability of our approach.

Table 9: **Ablation study on Stable Diffusion (SD) version.**

SD version	SAD↓	MSE↓	GRAD↓	CONN↓
SD v1.5	7.37	0.0264	6.55	5.31
SD v2.0	6.50	0.0232	6.72	4.98
SD v2.1	6.18	0.0219	6.42	4.70

6.5 Computational Complexity Comparison

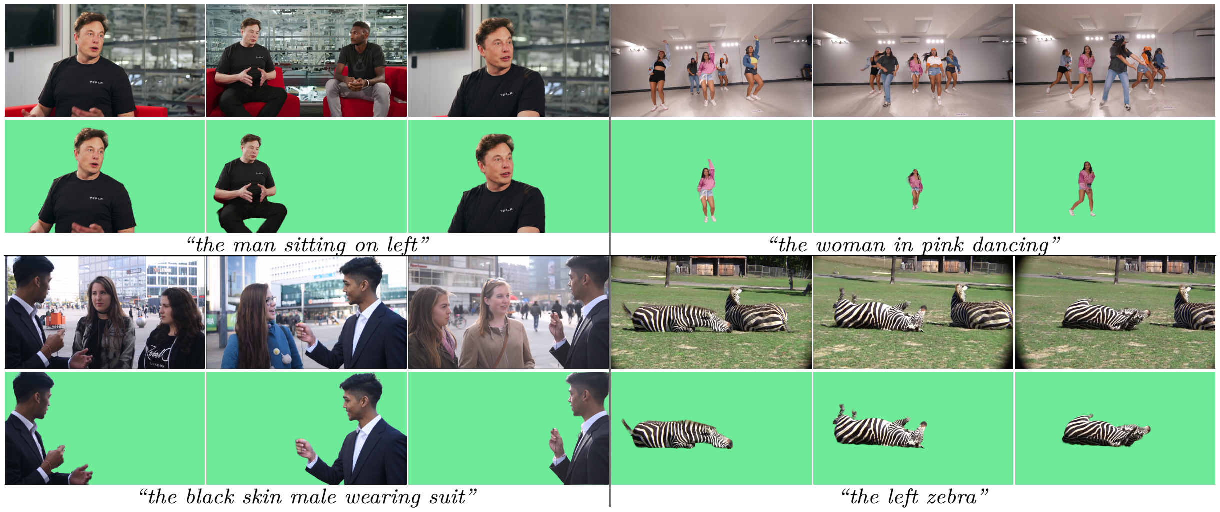
We compare the computational cost of our model with 4 baselines with competitive performance in Tab. 10 with number of parameters (#Params.), GMac, and GFlops as metrics. Our method has both the lowest model size and computational cost among them, demonstrating the efficiency of ours comprehensively.

Table 10: **Computational complexity comparison.**

	#Params.↓	GMACs↓	GFLOPs↓
PSALM+MGM	1617M	1430	2859
RefVPD+MGM	929M	959.4	1919
PSALM+MaGGIe	1617M	1429	2858
RefVPD+MaGGIe	930M	958.9	1918
Ours	**429M**	**769.7**	**1539**

6.6 Towards Video Instance Matting

Leveraging text prior enables us to easily extend our model to video instance matting, since the language is also capable of describing a series of frames. We show the qualitative result of our model on video instance matting in Fig. 9. Without any temporal modeling or fine-tuning, our model can still produce high-quality results on video sample with given text input. Our model is also robust to multi-instance environments and camera viewpoint changes, which demonstrates the powerful practicality of our model in real-world applications. More video instance matting results can also be found in the **supplementary video file**.



"the man sitting on left"

"the woman in pink dancing"

"the black skin male wearing suit"

"the left zebra"

Figure 9: **Qualitative results of our model on video instance matting.** The corresponding text inputs are shown in *italic* below every sample. The frames are placed following the temporal order in the original video from left to right.

6.7 More Qualitative Results

In Fig. 10, we show more qualitative comparison results on RefMatte-Test and RefMatte-RW100. Our model outperform other baselines both on synthetic and real-world datasets, demonstrating the superior performance of our model in grounding alpha matte from text descriptions. We also show more qualitative results on portrait and animal matting (Fig. 11). The text inputs are generated by BLIP2 (Li et al., 2023b). The results further validate the generalization ability of our model on different matting tasks.

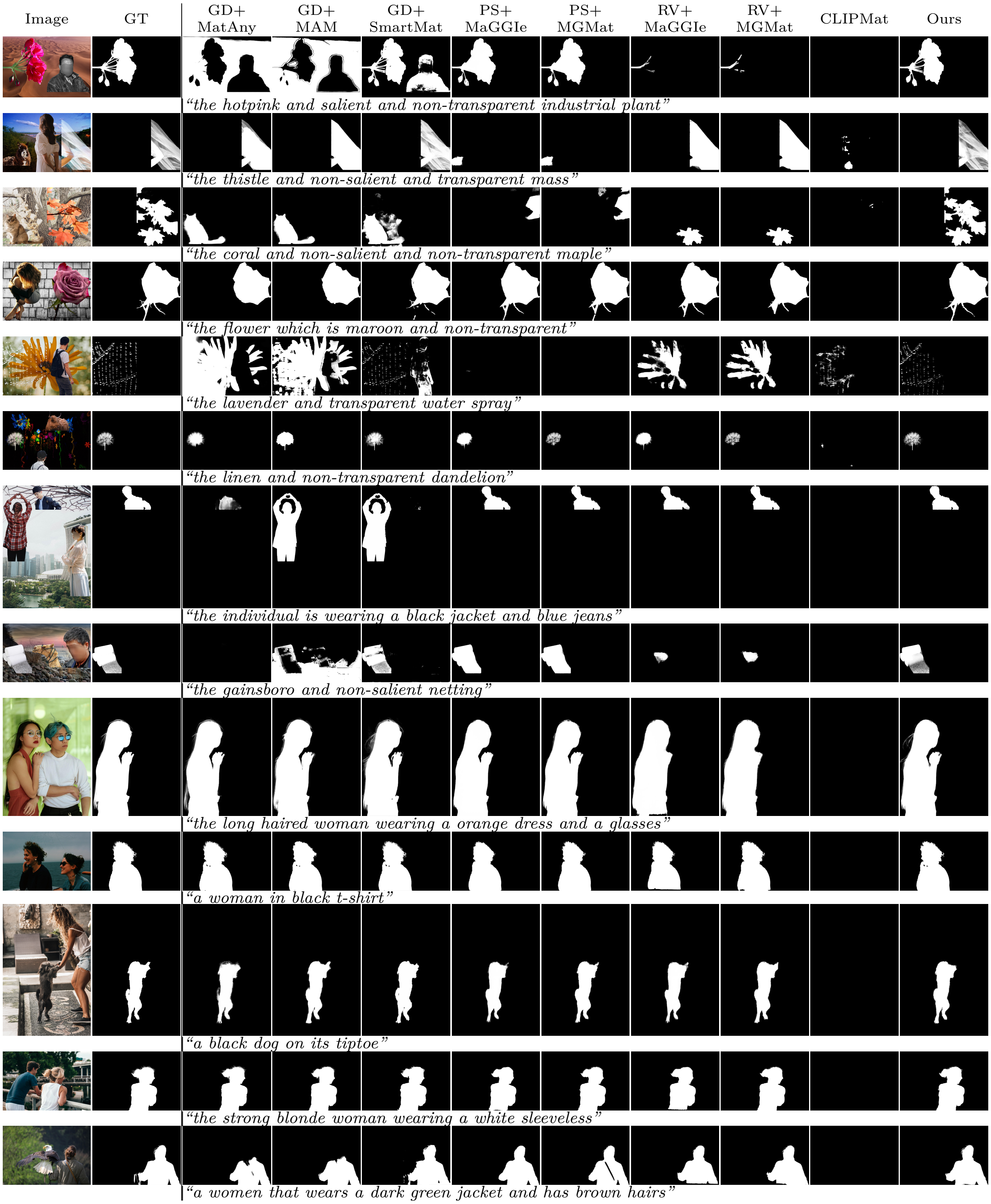
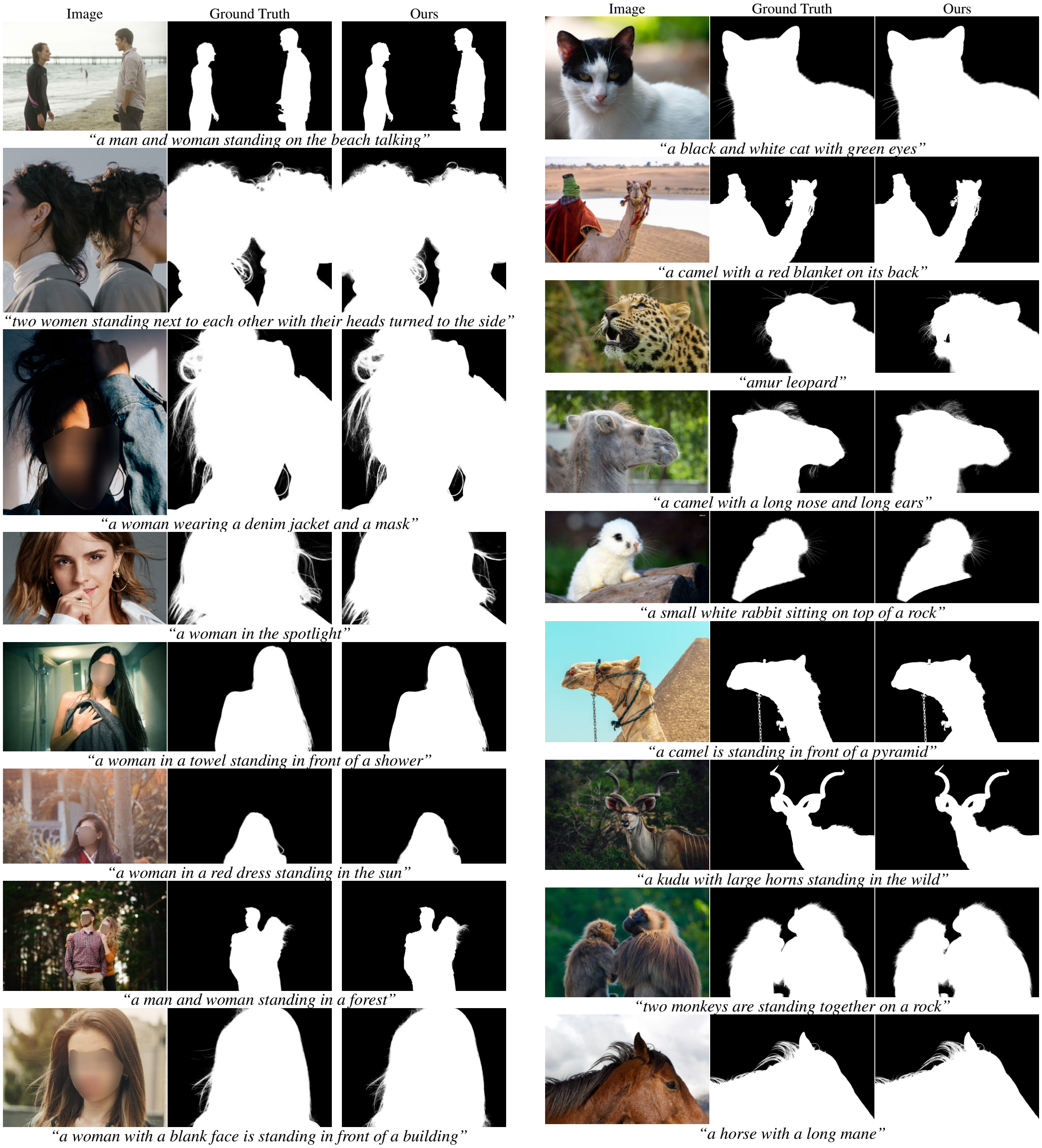


Image
GT
GD+ MatAny
GD+ MAM
GD+ SmartMat
PS+ MaGGIe
PS+ MGMat
RV+ MaGGIe
RV+ MGMat
CLIPMat
Ours
"the hotpink and salient and non-transparent industrial plant"
"the thistle and non-salient and transparent mass"
"the coral and non-salient and non-transparent maple"
"the flower which is maroon and non-transparent"
"the lavender and transparent water spray"
"the linen and non-transparent dandelion"
"the individual is wearing a black jacket and blue jeans"
"the gainsboro and non-salient netting"
"the long haired woman wearing a orange dress and a glasses"
"a woman in black t-shirt"
"a black dog on its tiptoe"
"the strong blonde woman wearing a white sleeveless"
"a women that wears a dark green jacket and has brown hairs"

Figure 10: **More qualitative comparisons on soft grounding task.** The text inputs are shown in *italic* below. Zoom in for better view.



"a man and woman standing on the beach talking"

"two women standing next to each other with their heads turned to the side"

"a woman wearing a denim jacket and a mask"

"a woman in the spotlight"

"a woman in a towel standing in front of a shower"

"a woman in a red dress standing in the sun"

"a man and woman standing in a forest"

"a woman with a blank face is standing in front of a building"

"a black and white cat with green eyes"

"a camel with a red blanket on its back"

"amur leopard"

"a camel with a long nose and long ears"

"a small white rabbit sitting on top of a rock"

"a camel is standing in front of a pyramid"

"a kudu with large horns standing in the wild"

"two monkeys are standing together on a rock"

"a horse with a long mane"

Figure 11: **Qualitative results of our model on portrait (left) and animal (right) matting.** Zoom in for better view.