Static-to-Dynamic: Animating Still Mattes via Generative Motion for Video Matting

Suqi Song[1], **Jiaqi Xu**[3], **Jialuo He**[1], **Renjing Pei**[3], **Lei Chen**[1,2], **Lei Zhu**[1,2*]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Huawei Noah's Ark Lab

Abstract

Video matting is essential for high-precision foreground extraction, yet its development is constrained by the scarcity of diverse annotated training data. To address this, we propose Static-to-Dynamic (S2D), a novel generative paradigm that scales video matting data by animating still labeled images into high-fidelity video-matte pairs. Starting from a static image-matte pair, S2D leverages an image-to-video generative model to synthesize realistic motion and scene dynamics, while simultaneously producing pseudo alpha labels from generative priors. Specifically, we introduce Latent Matte Propagation (LMP), which propagates the first-frame matte to subsequent generated frames via the attention mechanisms in the diffusion transformer. We further design a Coarse-to-Fine Matte Decoder to recover pixel-level alpha details from compressed latent representations using multi-scale generative features. Based on this paradigm, we build S2D-VM, a large-scale dataset containing 14K high-fidelity paired video clips. We also propose a Self-Corrected Progressive Training strategy to mitigate pseudo-label noise during optimization. Experiments show that S2D effectively scales video matting data and consistently improves the performance of existing models across multiple benchmarks.

1 Introduction

Video human matting estimates alpha mattes to precisely extract human foregrounds, making it fundamental for video editing. Despite recent architectural advances [6, 12, 2, 10, 4, 31], data remains the main bottleneck for robust matting systems. Unlike semantic segmentation, matting requires sub-pixel precision for fine details such as hair. High-quality annotations are difficult to obtain, which limits the scale and diversity of datasets. Existing datasets are often captured in green-screen studios using chroma keying to obtain ground truth [11, 34]. It depends on expensive equipment and controlled environments, and static backgrounds restrict scene diversity. To address this, the prevalent paradigm composites foreground objects onto random backgrounds [12, 10, 5, 31]. However, such composited videos often suffer from domain gaps from real-world footage in illumination, motion blur, and edge blending, which hinders model generalization in complex real scenarios.

The rapid evolution of generative models [14, 17, 15, 32, 7, 23] provides a promising avenue for alleviating data scarcity. Prior studies [22, 26, 13, 27, 29, 8] have demonstrated the feasibility of synthesizing image-label pairs for visual tasks, inspiring us to explore whether generative methods can expand video matting data. However, adapting existing generative paradigm for video matting faces several limitations. Previous work mainly focuses on image generation and typically derives pseudo-labels via text-visual cross-attention [13, 27]. In contrast, video generation requires more complex textual descriptions to capture temporal dynamics. As a result, text-driven methods struggle

*Lei Zhu is the corresponding author of this work.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

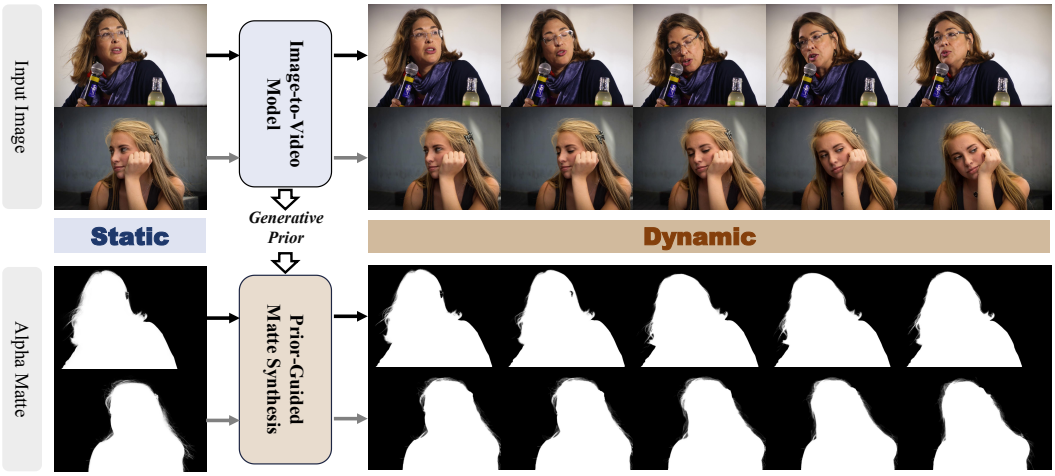


Figure 1: Overview of S2D paradigm. S2D transforms a labeled static image into a dynamic video clip and its corresponding matte sequence. The image-to-video branch introduces motion and temporal variation, while the prior-guided matte synthesis branch transfers the static annotation to generated frames, enabling scalable construction of video-matte pairs from image matting data.

to achieve precise object localization and temporal consistency. Moreover, existing works are largely confined to coarse segmentation tasks [13, 26, 27, 29]. It remains an open challenge how the internal features of generative models can be effectively mined to construct matting labels with precise details.

In this paper, we propose Static-to-Dynamic (S2D), a novel generative paradigm that transforms labeled static images into dynamic video-matte pairs. As illustrated in Fig. 1, given the abundance of annotated still images in image matting, S2D leverages advanced image-to-video models by treating a static image as the first frame and using textual descriptions to induce realistic motion, thereby synthesizing high-fidelity video sequences. Simultaneously, S2D extracts multi-level generative priors to construct pseudo labels in two steps. First, we introduce Latent Matte Propagation (LMP), an attention-based method applied within the Diffusion Transformer (DiT) during denoising. Taking the annotated first frame as an anchor, LMP leverages self-attention to establish RoPE-free affinities between the anchor and the generated frames. Combined with RoPE-conditioned self-affinity, these affinities enable temporal propagation of the anchor alpha matte to subsequent frames, yielding coarse latent mattes that provide reliable spatial localization for the synthesized content. Second, we employ a Coarse-to-Fine Matte Decoder to recover pixel-level alpha details from the compressed latent space. Built upon the coarse spatial structure encoded in the latent mattes, the decoder fuses multi-scale VAE decoder features from the image-to-video branch to exploit rich generative cues, assisting the matting branch in recovering intricate edge structures and subtle details.

Building on this paradigm, we construct S2D-VM, a large-scale video matting dataset generated from still image-matte pairs. Comprising 14K high-fidelity paired video clips, S2D-VM captures natural human motion, lighting variations, and diverse camera movements, effectively narrowing the domain gap between naive foreground-background composites and real-world scenes. To utilize this dataset for training video matting models despite the inevitable noise in pseudo labels, we further propose a Self-Corrected Progressive Training strategy. The model is first trained on conventional datasets with ground-truth labels to establish the basic matting ability, followed by joint training with the synthesized S2D-VM data to improve semantic stability and generalization in real-world scenes. Before progressing to more advanced stages, such as high-resolution or long-sequence training, the model from the current stage is utilized to refine and correct the generated labels, further improving boundary precision and temporal stability. Experiments on multiple benchmarks show that S2D scales video matting data and consistently boosts the performance of existing video matting models.

In summary, our main contributions are as follows:

- We provide a novel insight into scaling video matting data by proposing the Static-to-Dynamic (S2D) paradigm, which transforms the annotated static images into high-fidelity video-label pairs and serves as the foundation for our large-scale S2D-VM dataset.

- We develop a generative-prior-based matte labeling method that constructs pseudo labels via a two-step process, utilizing cross-frame latent affinities for temporal propagation and VAE decoder features to recover intricate edge structures.
- We introduce a Self-Corrected Progressive Training strategy that mitigates pseudo-label noise, enhances pseudo-label supervision through intermediate label self-correction, and improves model generalization when using generative video matting data.

2 Related Work

2.1 Video Matting

Video matting aims to estimate alphas for foreground extraction, with human matting being a major focus due to its broad applications. Existing approaches are generally divided into auxiliary-free and auxiliary-guided methods. Auxiliary-free models [6, 12, 2] predict mattes directly from raw frames, but often struggle with ambiguous backgrounds or visually similar distractors. Auxiliary-guided methods [10, 4, 5, 31, 30] leverage trimaps or masks to reduce task ambiguity and improve robustness. Despite these architectural advances, performance is heavily constrained by data scarcity. Existing datasets, such as VideoMatte240K [11] (484 videos) and VM800 [31] (826 videos), are limited in scale and rely on simple foreground-background composites that fail to bridge the real-world domain gap. To alleviate this, prior works [12, 10, 31] incorporate segmentation data, but this often leads to segmentation-like outputs with coarse boundaries [30]. Recent studies have explored various strategies for scaling high-quality matting data. MatAnyone 2 [30] introduces a matting quality evaluator to filter reliable training regions from noisy pseudo-labeled videos, while VideoMaMa [9] uses generative models to transform existing segmentation masks into alpha mattes. In contrast, we propose a new route for large-scale data acquisition: leveraging video generative models to animate abundant annotated images and jointly produce high-fidelity video sequences and alpha annotations.

2.2 Synthetic Data for Visual Tasks

Recent advances in generative models [14, 15, 17, 32, 23, 7] serve as a powerful synthesis engine for visual tasks, providing a virtually limitless resource of high-fidelity images and videos. This paradigm has been widely adopted to address specific data bottlenecks in several vision tasks. For instance, it is effectively utilized for large-scale classification pre-training [22] and for mitigating distribution imbalances in long-tail recognition [18]. In semantic segmentation, generative data synthesis is primarily categorized into image-to-mask and mask-to-image methodologies. Image-to-mask approaches, such as DiffuMask [27] and Dataset-Diffusion [13], extract semantic labels directly from internal generative features or cross-attention maps. Conversely, mask-to-image methods, including FreestyleNet [28] and FreeMask [29], synthesize high-fidelity images conditioned on specified semantic layouts. Beyond generation, training-free selection methods [21] have been introduced to identify the most beneficial synthetic samples for downstream segmenters. Despite these advancements, existing research remains predominantly limited to static image-label generation, and the potential of generative synthesis remains largely untapped in video matting.

3 Methodology

The core of our work involves mining generative priors to construct synchronized labels and investigating effective strategies for using synthesized data for improved video matting performance. In the following, we first detail our two-stage label construction method comprising Latent Matte Propagation in Section 3.1 and a Coarse-to-Fine Matte Decoder in Section 3.2. We then present the S2D-VM dataset in Section 3.3 and the Self-Corrected Progressive Training strategy in Section 3.4.

3.1 Latent Matte Propagation

As shown in Fig. 2, we build our text-image-to-video pipeline on Wan2.2 [23]. Given a reference image $I_0 \in \mathbb{R}^{H\times W\times c}$ and a text prompt generated by Qwen-VL [24], Wan synthesizes a video conditioned on both visual and textual inputs. Specifically, I_0 is encoded via VAE into an anchor latent $z_0 \in \mathbb{R}^{H'\times W'\times d}$, while the remaining latents $\{z_t\}_{t=1}^{T'}$ to be generated are initialized from

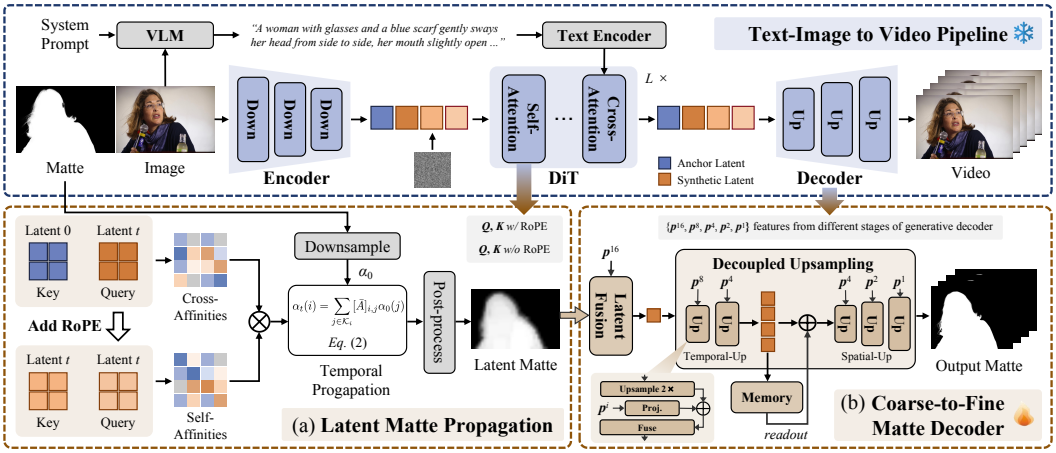


Figure 2: Architecture of S2D framework for video-matte generation. S2D establishes a text-image to video pipeline to animate a static image-matte pair based on motion prompts. The video label generation process consists of two core components: (a) Latent Matte Propagation (LMP), which extracts attention-based affinities from the DiT backbone to facilitate accurate matte transfer across frames in the latent space; and (b) a Coarse-to-Fine Matte Decoder, which fuses the propagated latent matte with multi-scale VAE decoder features to recover high-resolution alpha mattes.

Gaussian noise. The concatenated latent sequence $\mathcal{Z} = \{z_t\}_{t=0}^{T'}$ is iteratively denoised by the DiT and finally decoded into a video $\mathcal{V} \in \mathbb{R}^{(1+T)\times H\times W}$. During denoising, DiT self-attention naturally establishes dense structural and motion correspondences between the clean anchor z_0 and generated latents. Building on this observation, we introduce Latent Matte Propagation (LMP) to propagate the anchor alpha matte in latent space.

Concretely, LMP exploits DiT self-attention to establish dense correspondences between the anchor latent z_0 and each target latent z_t. At denoising step $s \in S$ and self-attention layer $l \in L$, we extract the query $Q_t^{(s,l)}$ from z_t and the key $K_0^{(s,l)}$ from z_0, following the default DiT attention formulation with Rotary Positional Embeddings [19] (RoPE). We then compute the ensemble cross-affinity $\bar{A}_{t\leftarrow 0} \in [0,1]^{H'W'\times H'W'}$ by averaging attention maps over denoising steps and layers:

$$\bar{A}_{t\leftarrow 0} = \frac{1}{|S|\cdot|L|}\sum_{s\in S}\sum_{l\in L}\text{softmax}\left(\frac{Q_t^{(s,l)}(K_0^{(s,l)})^\top}{\sqrt{d}}\right). \tag{1}$$

The latent matte $\alpha_t \in [0,1]^{H'\times W'}$ for z_t is then propagated from the downsampled anchor matte α_0 using the affinity $\bar{A}_{t\leftarrow 0}$ with a Top-K sparsification operator:

$$\alpha_t(i) = \sum_{j\in\mathcal{K}_i}[\bar{A}]_{i,j}\alpha_0(j). \tag{2}$$

Intuitively, the alpha value of the i-th target token is computed as a weighted aggregation of the anchor matte values $\alpha_0(j)$ over the top-k most similar locations in the anchor frame.

RoPE-Free Cross-Affinity. A key observation is that the default use of RoPE in DiT attention is not always optimal for cross-latent matte propagation. While RoPE improves spatial reasoning, it also biases attention toward tokens with similar coordinates. For cross-frame correspondence under object motion, this positional bias becomes less suitable. As illustrated in Fig. 3, when a foreground region in the anchor frame moves away and its original location becomes background in later generated frames, RoPE-based affinity may still assign a high response to this coordinate-aligned but semantically mismatched region. As a result, foreground alpha values are incorrectly propagated to the background, producing *ghosting artifacts*. Motivated by this observation, we remove RoPE from the queries and keys when computing cross-latent affinities, yielding a RoPE-free affinity $\bar{A}^{RF}_{t\leftarrow 0}$

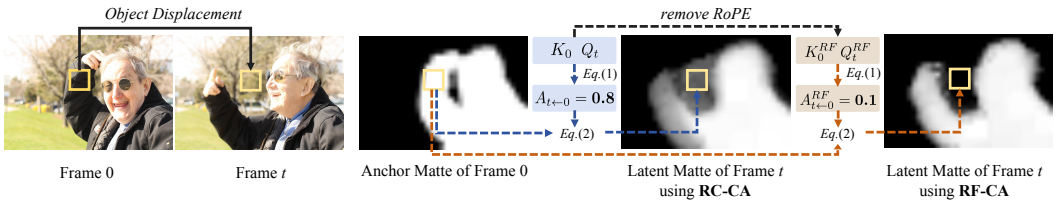


Figure 3: Comparison between RoPE-Conditioned Cross-Affinity (RC-CA) and RoPE-Free Cross-Afiinity (RF-CA). RF-CA effectively eliminates the ghosting artifacts introduced by RC-CA.

following Eq. (1). This simple modification makes the correspondence more content-driven and effectively suppresses coordinate-induced alpha leakage.

RoPE-Conditioned Self-Affinity. However, removing RoPE also weakens spatial constraints that suppress false matches between visually similar foreground and background tokens. This makes propagation more vulnerable to semantic ambiguities, causing background noise and foreground holes. To restore spatial regularization, we introduce a RoPE-conditioned self-affinity $\bar{A}_{t\leftarrow t}$, computed from self-attention within z_t while preserving positional embeddings:

$$\bar{A}_{t\leftarrow t} = \frac{1}{|S|\cdot|L|}\sum_{s\in S}\sum_{l\in L}\text{softmax}\left(\frac{Q_t^{(s,l)}(K_t^{(s,l)})^\top}{\sqrt{d}}\right). \tag{3}$$

The final refined affinity is then formulated with a temperature hyperparameter τ as:

$$A^*_{t\leftarrow 0} = (\bar{A}_{t\leftarrow t})^\tau \bar{A}^{RF}_{t\leftarrow 0}. \tag{4}$$

The latent alpha matte α_t is then obtained by applying $A^*_{t\leftarrow 0}$ in Eq. (2), followed by post-processing with threshold truncation and value stretching to suppress low-confidence noise and enhance contrast.

3.2 Coarse-to-Fine Matte Decoder

In the Wan model, pixel-space inputs are encoded into highly compressed latent representations of size $[1+T/4, H/16, W/16]$. Within this latent space, our training-free LMP generates latent mattes for synthesized frames, providing reliable target localization. However, due to severe spatio-temporal compression, the propagated mattes preserve only coarse object geometry and motion trajectories. Since the fine-grained alpha details lost in the latent space cannot be recovered by simple interpolation, we introduce a coarse-to-fine matte decoder, which lifts the coarse latent mattes to pixel space and leverages multi-scale generative features to reconstruct high-resolution alpha mattes.

Architectural Design. As shown in Fig. 2 (b), the matte decoder consists of a latent fusion block, decoupled up-sampling modules, and a memory module. We use multi-scale VAE decoder features $P = \{p^{16}, p^8, p^4, p^2, p^1\}$ from the image-to-video branch as generative priors, where p^i denotes the feature map at stride i. The decoder starts by fusing the latent matte α_t with p^{16} to anchor the predicted motion. Decoupled up-sampling then restores temporal resolution first, followed by spatial details. In the temporal stage, the decoder integrates p^8 and p^4 to expand the compressed temporal axis. At the bottleneck, an attention-based memory module retrieves long-range temporal context to enforce global smoothness. Finally, the spatial up-sampling stage progressively fuses p^4, p^2, and p^1 to reconstruct the output matte with pixel-level resolution and fine boundary details.

Model Training. We train the matte decoder using existing image and video matting datasets. Since these samples are not produced by the video generation model, their latent mattes can only be approximated. Specifically, we use the clean latents $\mathcal{Z}$ extracted by the VAE encoder as surrogates for the denoised generation latents, and feed them into the VAE decoder to obtain the multi-scale features P. We then construct latent mattes with data-specific strategies. For images, we downsample the ground-truth alphas and apply random morphological perturbations, including dilation, erosion and random noise, to mimic inference-stage artifacts. For videos, we construct latent mattes by

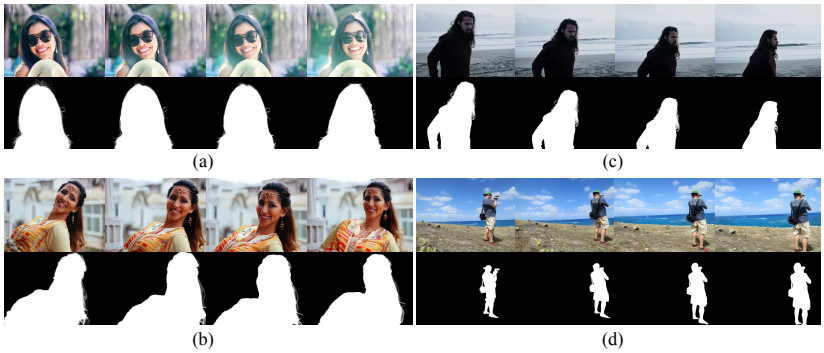


Figure 4: Samples from our constructed S2D-VM dataset. (a)(b) show natural illumination variations and human motion, while (c)(d) illustrate diverse camera movements, such as zoom in and zoom out.

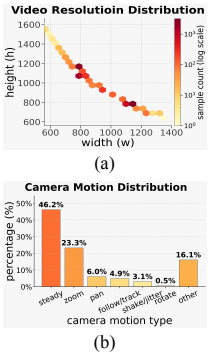


Figure 5: Distribution statistics of S2D-VM.

adding Gaussian noise to $\mathcal{Z}$ and applying the same latent matte propagation process within DiT. This improves distribution consistency between training and actual generation.

3.3 S2D-VM Dataset

Following the S2D pipeline in Fig. 2, we construct the S2D-VM dataset from three image matting sources: HHM50K [20], RWP-636 [33], and PPM-100 [6]. We filter out images with small portrait areas, insufficient resolution, or redundant scenes. The remaining images serve as references, while Qwen-VL [24] generates descriptive prompts covering object motion, scene transitions, and camera trajectories. The S2D framework then produces videos and labels, with examples shown in Fig. 4.

Dataset Statistics. S2D-VM contains 14K video clips and 1M frames, providing a substantial scale-up over existing synthetic datasets such as VM240K [11] with 484 videos and 240K frames. During generation, the video resolution is maximized to a total area not exceeding 1280×704 pixels while preserving the original aspect ratio of the source image. As shown in Fig. 5, in contrast to the fixed resolution of VM240K, the S2D-VM dataset provides diverse aspect ratios and thus covers a broader range of visual compositions. In addition, S2D-VM includes varied camera motions guided by descriptive prompts, such as steady, zoom, pan, follow, shake, and rotate. These motions go beyond conventional rigid trajectories and provide richer dynamic environments.

3.4 Self-Corrected Progressive Training

We train existing video matting models with S2D-VM. Results (Table 4) show that training on S2D-VM substantially improves performance compared with training on a single video matting dataset, yet the unavoidable noise in pseudo labels still limits further gains. To better exploit the synthetic data, we propose a self-corrected progressive training strategy that improves model robustness through clean supervision, noisy-data learning, and label self-correction.

We adopt a standard two-stage optimization scheme, progressing from low-resolution short-sequence training to high-resolution long-sequence refinement. In the first stage, the model is trained on VideoMatte240K [11] to establish basic matting ability, and then jointly trained with S2D-VM to improve generalization. The low-resolution, short-sequence setting naturally reduces the impact of pseudo-label noise. In the second stage, however, high-resolution long-sequence training amplifies temporal instability and boundary noise, making further optimization more challenging.

Label Self-Correction. Before the second stage, we refine the synthetic labels based on first-frame quality and temporal stability. For a synthetic video, let $\mathcal{O} = \{O_0, \ldots, O_T\}$ denote the original synthesized labels and $\mathcal{N} = \{N_0, \ldots, N_T\}$ the labels predicted by the first-stage matting model. We first assess the quality of O_0 and N_0 against the ground truth G_0 using MAD and Grad [16], and define the gain score $Gain$ as the error reduction from the original label to the new one. If the new label is clearly worse ($Gain < -g$), we keep $\mathcal{O}$; if it is clearly better ($Gain > g$), we replace the entire sequence with $\mathcal{N}$. For ambiguous cases, we perform a region-wise update. Specifically, we

Table 1: Quantitative comparisons of different video matting methods across multiple video matting benchmarks. Gray highlights indicate better results of the S2D-optimized models compared with the corresponding baseline models. **Bold** numbers denote the best results among all compared methods.

Metrics	Auxiliary-free Methods				Mask-guided Methods			
	GVM [2]	RVM [12]	RVM-Large [12]	RVM-Large **+S2D**	FTP-VM [4]	MaGGie [5]	MatAnyone [31]	MatAnyone **+S2D**
VideoMatte (512×288)								
MAD ↓	6.39	6.08	5.32	5.06	6.13	5.49	5.15	**4.92**
MSE ↓	1.82	1.47	0.62	0.62	1.31	**0.60**	0.93	0.78
Grad ↓	0.72	0.88	0.59	0.57	1.14	0.57	0.67	**0.56**
dtSSD ↓	1.27	1.36	1.24	1.20	1.60	1.39	1.18	**1.12**
Conn ↓	0.43	0.41	0.30	0.25	0.41	0.31	0.26	**0.22**
VideoMatte (1920×1080)								
MAD ↓	6.33	6.57	5.81	5.77	8.00	4.42	4.24	**4.24**
MSE ↓	2.08	1.93	0.97	0.95	3.24	0.40	0.33	**0.32**
Grad ↓	8.04	10.55	9.65	9.38	23.75	4.03	4.00	**3.83**
dtSSD ↓	1.59	1.90	1.78	1.73	2.37	1.31	1.19	**1.19**
YoutubeMatte (512×288)								
MAD ↓	3.30	4.08	3.36	2.82	3.08	3.54	2.72	**2.33**
MSE ↓	1.52	1.97	1.04	1.01	1.29	1.23	1.01	**0.83**
Grad ↓	0.79	1.34	1.03	0.94	1.16	1.10	0.97	**0.77**
dtSSD ↓	1.52	1.81	1.62	1.52	1.83	1.88	1.60	**1.45**
Conn ↓	0.44	0.60	0.50	0.43	0.41	0.49	0.39	**0.32**
YoutubeMatte (1920×1080)								
MAD ↓	2.68	4.37	3.58	3.10	6.49	2.37	1.99	**1.54**
MSE ↓	1.34	2.25	1.23	1.19	4.58	0.98	0.71	**0.47**
Grad ↓	8.38	15.10	12.97	11.78	29.78	7.69	8.91	**5.54**
dtSSD ↓	1.72	2.28	2.04	1.92	2.41	1.77	1.65	**1.42**
CRGNN (1920×1080)								
MAD ↓	5.03	5.98	5.75	5.37	6.64	9.50	5.76	**4.87**
MSE ↓	**2.15**	2.79	2.47	2.44	3.54	6.11	3.04	2.38
Grad ↓	14.28	13.68	13.26	**12.85**	15.10	16.51	15.55	13.65
dtSSD ↓	**4.86**	5.36	5.17	4.97	5.98	6.02	5.44	5.02

partition the first frame according to G_0 into foreground ($\alpha = 1$), background ($\alpha = 0$), and a slightly dilated transition region ($0 < \alpha < 1$). A region is updated only when its local gain $Gain_r$ exceeds the threshold g. To avoid temporal artifacts, we further require the new labels to have better temporal consistency than the original labels. The refined labels are then used for second-stage training.

4 Experiments

4.1 Implementation details

We use Wan2.2 TI2V-5B for video generation and Qwen-VL-7B for prompt synthesis. In latent matte propagation, we set the top-k ratio to 0.07 and $\tau = 1$, and compute cross- and self-affinities by averaging over 50 timesteps and 30 transformer layers. The matte decoder is trained on VM240K [11] and HHM50K [20] for 25 epochs in total, using the AdamW optimizer with a learning rate of 1×10^{-4}. We adopt the same training losses as prior work [12]. We further use S2D-VM to optimize existing matting models, including RVM-Large [12] and MatAnyone [31]. During training, we keep the original architectures, losses, and optimizer settings unchanged, but replace their original multi-source training data with S2D-VM and VM240K under our proposed self-correction progressive training strategy.

4.2 Comparisons

The proposed S2D integrates a novel data generation paradigm, the S2D-VM dataset, and a specialized training strategy. To validate its efficacy, we optimize two baseline models, RVM-Large [12] and MatAnyone [31], using our dataset and training protocol. We then compare these S2D-optimized versions against existing state-of-the-art methods, including *auxiliary-free* methods: RVM [12], RVM-Large [12], GVM [2]; and *mask-guided* methods: FTP-VM [4], Maggie [5] and MatAnyone [31].



Figure 6: Qualitative comparisons on real-world videos. Applying S2D to both the *auxiliary-free* RVM-Large [12] and the *mask-guided* MatAnyone [31] leads to improved matting results over the original versions, with more stable tracking of moving objects and better preservation of fine details.

Table 2: Ablation study of core designs in latent matte propagation. RC/RF denote RoPE-conditioned/RoPE-free, and CA/SA denote cross-affinity/self-affinity, respectively. The best results are in **bold**.

RC-CA	RF-CA	RC-SA	Post-Process	VideoMatte		YoutubeMatte	
				MSE ↓	MAD ↓	MSE ↓	MAD ↓
✓				10.43	59.35	14.77	79.31
	✓			7.29	42.96	9.39	57.84
	✓	✓		6.56	28.43	6.95	31.93
	✓	✓	✓	**6.05**	**26.19**	**5.99**	**24.66**

Table 3: Ablation study of key components in matte decoder. LM and MSF denote latent matte and multi-scale features, respectively. The best results are in **bold**.

Method	VideoMatte			
	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
Matte Decoder	**4.90**	**0.69**	**5.36**	**1.54**
w/o LM-Fusion	6.60	2.07	8.73	1.76
w/o MSF-Fusion	7.79	3.31	10.08	1.82
w/o Memory Module	6.35	1.99	7.69	1.79

4.2.1 Quantitative Results

Following prior works [12, 31], we evaluate on two synthetic benchmarks (VideoMatte [12] and YoutubeMatte [31]) and a real-world benchmark (CRGNN [25]). We use standard metrics, including MAD and MSE for overall accuracy, Grad [16] for detail fidelity, Conn [16] for connectivity quality, and dtSSD [1] for temporal consistency.

As shown in Table 1, S2D consistently improves both RVM-Large [12] and MatAnyone [31] across all benchmarks and metrics, demonstrating strong model-agnosticity and validating the effectiveness of the proposed data and training strategy. The improvements are particularly evident on YouTubeMatte, where MatAnyone**+S2D** yields clear gains in MAD, Grad, and dtSSD, indicating better overall matting quality, finer boundary details, and stronger temporal consistency. For instance, on YouTubeMatte (1920 $\times$ 1080), MAD drops from 1.99 to 1.54, Grad from 8.91 to 5.54, and dtSSD from 1.65 to 1.42. Similar improvements also transfer to the real-world benchmark CRGNN, where MatAnyone**+S2D** improves MAD from 5.76 to 4.87, Grad from 15.55 to 13.65, and dtSSD from 5.44 to 5.02. Moreover, MatAnyone**+S2D** establishes new best results on the majority of benchmarks. On the real-world dataset CRGNN, its performance is also highly competitive with the diffusion-based GVM [2].

4.2.2 Qualitative Results

Qualitative comparisons on real-world videos are shown in Fig. 6. Benefiting from the more diverse motion patterns introduced in S2D-VM dataset, the models trained with S2D exhibit more stable

Table 4: Effect of training data and protocol on YoutubeMatte. VM and Seg represent VM240K [11] and segmentation datasets, respectively. SC-PT denotes our training strategy.

Training Setting	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
(a) VM	13.52	11.80	10.49	2.21
(b) VM + Seg	5.21	3.12	9.32	1.70
(c) S2D-VM	2.30	1.03	9.72	1.79
(d) VM + S2D-VM	2.19	0.83	7.99	1.65
(e) VM + S2D-VM *w/* **SC-PT**	**1.54**	**0.47**	**5.54**	**1.42**

Table 5: Effect of synthetic data scale on YoutubeMatte, including 2K, 4K, 8K and 14K, with 14K corresponding to the full dataset scale.

Data Scale	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
2K	3.62	2.30	11.56	2.16
4K	3.29	1.91	10.04	1.91
8K	2.68	1.36	10.06	1.85
14K	**2.30**	**1.03**	**9.72**	**1.79**

tracking of moving objects. For instance, in the first-column case, the S2D-optimized models produce more accurate and stable predictions for the motion-blurred hand, while the original models are more prone to missing foreground regions. Meanwhile, fine structures such as hair strands are also better preserved, leading to cleaner boundaries and more complete object shapes. These results further support the effectiveness of S2D for improving both motion robustness and fine-detail matting quality.

4.3 Ablation Studies

Core Designs in LMP. We evaluate the proposed latent propagation strategy on VideoMatte and YouTubeMatte. Since these test samples are not generated, we adopt the approximate offline construction method introduced in Section 3.2 to derive latent mattes. We use different propagation strategies and compare them with downsampled ground-truth mattes. Results are reported in Table 2. Comparing the two cross-affinity variants, RoPE-free cross-affinity (RF-CA) consistently outperforms RoPE-conditioned cross-affinity (RC-CA) on both datasets, suggesting that removing RoPE-induced positional bias benefits cross-latent matching. Building on RF-CA, adding RoPE-conditioned self-affinity (RC-SA) brings further gains, indicating that intra-latent spatial constraints effectively complement cross-latent semantic matching. Finally, post-processing delivers the best performance.

Key Components in Matte Decoder. Table 3 reports the ablation results of the key components in the matte decoder. Based on the full model, we separately remove the Latent Matte (LM) from latent fusion block, remove the Multi-Scale Features (MSF) from the up-sampling modules, and remove the memory module to assess their contributions. All three components are beneficial, and the complete model achieves the best performance across all metrics. Specifically, removing LM-Fusion increases MAD/MSE from 4.90/0.69 to 6.60/2.07, showing that the latent matte provides important coarse structural guidance during decoding. Removing MSF-Fusion causes a larger drop, which highlights the importance of multi-scale features for recovering fine boundaries and temporal structures.

Effect of Training Data and Protocol. We ablate the effect of training data and protocol on MatAnyone [31], with results in Table 4. Training on VM240K [11] alone leads to the worst performance, while adding segmentation data yields an improvement. Notably, training on S2D-VM alone significantly outperforms setting (a), and surpasses the setting (b) in terms of MAD and MSE, highlighting the high quality of the proposed dataset. Directly combining S2D-VM with VM240K further improves the results. Building on setting (d), the proposed SC-PT brings additional gains by mitigating label noise, leading to the best performance across all metrics.

Effect of Data Scale. We study the effect of synthetic data scale by randomly sampling subsets of different sizes from S2D-VM and training MatAnyone [31] on each subset. As shown in Table 5, performance improves consistently as the data scale increases. In particular, scaling from 2K to 14K reduces MAD from 3.62 to 2.30 and MSE from 2.30 to 1.03. The gain is especially pronounced from 4K to 8K, and becomes more moderate thereafter. These results suggest that S2D-VM provides effective supervision for video matting and continues to benefit performance as its scale grows.

5 Conclusion

We presented Static-to-Dynamic (S2D), a novel generative paradigm for scaling video matting data by transforming labeled still images into high-fidelity video-matte pairs. By leveraging generative priors from an image-to-video model, S2D enables temporally coherent matte construction through Latent

Matte Propagation and recovers fine-grained alpha details with a Coarse-to-Fine Matte Decoder. Built upon this paradigm, we further constructed S2D-VM, a large-scale synthetic video matting dataset containing 14K paired clips, and introduced a Self-Corrected Progressive Training strategy to effectively exploit noisy pseudo labels during optimization. Extensive experiments demonstrate that S2D consistently improves existing video matting models across multiple benchmarks, validating the effectiveness of generative data scaling for this task.

Acknowledgments and Disclosure of Funding

This work is supported by Guangdong Basic and Applied Basic Research Foundation (2025A1515140021), Guangdong Science and Technology Department (2024ZDZX2004), and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things(No.2023B1212010007).

References

[1] Mikhail Erofeev, Yury Gitman, Dmitriy S Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, volume 1, page 2, 2015.

[2] Yongtao Ge, Kangyang Xie, Guangkai Xu, Li Ke, Mingyu Liu, Longtao Huang, Hui Xue, Hao Chen, and Chunhua Shen. Generative video matting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025.

[3] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[4] Wei-Lun Huang and Ming-Sui Lee. End-to-end video matting with trimap propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14337–14347, 2023.

[5] Chuong Huynh, Seoung Wug Oh, Abhinav Shrivastava, and Joon-Young Lee. Maggie: Masked guided gradual human instance matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3870–3879, 2024.

[6] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 1140–1147, 2022.

[7] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[8] Yuhang Li, Xin Dong, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. A simple background augmentation method for object detection with diffusion model. In *European Conference on Computer Vision*, pages 462–479. Springer, 2024.

[9] Sangbeom Lim, Seoung Wug Oh, Jiahui Huang, Heeji Yoon, Seungryong Kim, and Joon-Young Lee. Videomama: Mask-guided video matting via generative prior. *arXiv preprint arXiv:2601.14255*, 2026.

[10] Chung-Ching Lin, Jiang Wang, Kun Luo, Kevin Lin, Linjie Li, Lijuan Wang, and Zicheng Liu. Adaptive human matting for dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10229–10238, 2023.

[11] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8762–8771, 2021.

[12] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022.

[13] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *Advances in Neural Information Processing Systems*, 36:76872–76892, 2023.

[14] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[15] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[16] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1826–1833. IEEE, 2009.

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[18] Joonghyuk Shin, Minguk Kang, and Jaesik Park. Fill-up: Balancing long-tailed data with generative models. *arXiv preprint arXiv:2306.07200*, 2023.

[19] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[20] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Ultrahigh resolution image/video matting with spatio-temporal sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14112–14121, 2023.

[21] Hao Tang, Siyue Yu, Jian Pang, and Bingfeng Zhang. A training-free synthetic data selection method for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7229–7237, 2025.

[22] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36:48382–48402, 2023.

[23] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[24] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[25] Tiantian Wang, Sifei Liu, Yapeng Tian, Kai Li, and Ming-Hsuan Yang. Video matting via consistency-regularized graph neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4902–4911, 2021.

[26] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems*, 36:54683–54695, 2023.

[27] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023.

[28] Han Xue, Zhiwu Huang, Qianru Sun, Li Song, and Wenjun Zhang. Freestyle layout-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14256–14266, 2023.

[29] Lihe Yang, Xiaogang Xu, Bingyi Kang, Yinghuan Shi, and Hengshuang Zhao. Freemask: Synthetic images with dense annotations make stronger segmentation models. *Advances in Neural Information Processing Systems*, 36:18659–18675, 2023.

[30] Peiqing Yang, Shangchen Zhou, Kai Hao, and Qingyi Tao. Matanyone 2: Scaling video matting via a learned quality evaluator. *arXiv preprint arXiv:2512.11782*, 2025.

[31] Peiqing Yang, Shangchen Zhou, Jixin Zhao, Qingyi Tao, and Chen Change Loy. Matanyone: Stable video matting with consistent memory propagation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7299–7308, 2025.

[32] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[33] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1154–1163, 2021.

[34] Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, Qixing Huang, and Weiwei Xu. Attention-guided temporally coherent video object matting. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5128–5137, 2021.

A Appendix A

A.1 Implementation Details.

Affinity Refinement Mechanism. Let each frame contain N latent tokens. The RoPE-free cross-affinity $\bar{A}^{RF}_{t\leftarrow 0} \in \mathbb{R}^{N\times N}$ measures target-to-anchor correspondence, while the RoPE-conditioned self-affinity $\bar{A}_{t\leftarrow t} \in \mathbb{R}^{N\times N}$ captures target-frame internal structure. Equation (4) combines them as

$$A^{*}_{t\leftarrow 0} = (\bar{A}_{t\leftarrow t})^{\tau}\bar{A}^{RF}_{t\leftarrow 0}. \tag{5}$$

For an entry in the resulting affinity, we have

$$A_{t\leftarrow 0}(i,k)^{*} = \sum_{j}(\bar{A}_{t\leftarrow t})^{\tau}(i,j)\bar{A}^{RF}_{t\leftarrow 0}(j,k). \tag{6}$$

Here, $(\bar{A}_{t\leftarrow t})^{\tau}(i,:)$ represents the self-affinity distribution between target token i and all target-frame tokens, indicating which target-frame tokens are structurally related to i. In contrast, $\bar{A}^{RF}_{t\leftarrow 0}(:,k)$ represents the cross-affinity between all target-frame tokens and anchor token k, indicating which target-frame tokens match anchor token k. Therefore, their inner product can be interpreted as a structure-aware weighted vote: target-frame tokens structurally related to i vote for whether target token i should correspond to anchor token k.

This mechanism explains why Eq. (4) reduces background noise and foreground holes. Specifically, if a background token i visually resembles an anchor foreground token k, RF-CA may produce a high direct response $\bar{A}^{RF}_{t\leftarrow 0}(i,k)$. However, when this is an isolated mismatch, the target-frame tokens selected by $\bar{A}_{t\leftarrow t}(i,:)$ usually do not consistently match the same anchor foreground token, so the weighted vote suppresses this response. Conversely, if a true foreground token i has a weak RF-CA response, its structurally related foreground tokens may still have strong correspondence to anchor foreground tokens. Their support is aggregated through Eq. (4), increasing the final affinity of token i and alleviating foreground holes.

Moreover, the parameter τ controls how strongly the RC-SA prior affects the refined affinity. When $\tau = 1$, the original, unscaled self-affinity is used; when $\tau > 1$, the self-affinity distribution becomes sharper, so the propagation relies more on high-confidence target-frame structural relations. Thus, τ is retained as a strength-control parameter for cases where stronger self-affinity regularization may be needed.

Post-Processing of LMP. After obtaining the propagated latent matte via Eq. (2), we further apply a simple post-processing procedure to suppress residual noise and reduce the distribution mismatch with the ground-truth alpha of the first frame. The procedure includes thresholding, local consistency refinement, and adaptive stretching.

We first perform confidence thresholding on the propagated matte α to remove low-confidence responses and obtain α^{th}. Specifically, we use the minimum non-zero value of the first-frame ground-truth alpha as the threshold, and suppress responses below this value to zero. This step helps eliminate weak activations caused by unreliable propagation. We then apply local consistency refinement based on neighborhood statistics within a local window. For each location, we compute its local average response and local maximum response, and update the matte according to the local density level:

$$\alpha_i^{\mathrm{ref}} = \begin{cases} 0, & \mu_i \le t_{\mathrm{low}}, \\ \alpha_i^{\mathrm{th}}, & t_{\mathrm{low}} < \mu_i \le t_{\mathrm{high}}, \\ m_i, & \mu_i > t_{\mathrm{high}}, \end{cases} \tag{7}$$

where μ_i and m_i denote the local average and local maximum response around location i, respectively, and t_{low} and t_{high} are two thresholds. This step suppresses isolated sparse noise while strengthening spatially consistent regions. Finally, we apply value stretching over the non-zero region to improve contrast and better align the propagated matte distribution with that of the anchor alpha:

$$\alpha_i^{\mathrm{str}} = \begin{cases} \dfrac{\alpha_i^{\mathrm{ref}} - a_{\min}}{a_{\max} - a_{\min} + \delta}(1 - a_{\min}) + a_{\min}, & i \in \mathcal{M}, \\ 0, & i \notin \mathcal{M}, \end{cases} \tag{8}$$

where $\mathcal{M} = \{i \mid \alpha_i^{\mathrm{ref}} > 0\}$ denotes the valid region, $a_{\min}$ and $a_{\max}$ are the minimum and maximum values within $\mathcal{M}$, and δ is a small constant for numerical stability. This step enhances contrast within the valid matte region while preserving zero-valued background locations.

Table 6: Effect of different attention layer ranges used in latent matte propagation on VideoMatte.

Layer Range	1–10	11–20	21–30	1–20	11–30	1–30
MSE ↓	8.20	9.67	14.49	8.44	8.20	**7.45**
MAD ↓	41.54	42.19	74.66	**39.00**	44.07	39.85

Table 7: Effect of different attention timestep ranges used in latent matte propagation on VideoMatte.

Step Range	1–10	11–20	21–30	31–40	41–50	11–30	21–40	31–50	11–40	21–50	11–50	1–50
MSE ↓	18.27	9.78	8.50	8.26	10.04	8.60	8.04	8.29	7.94	7.79	7.53	**7.45**
MAD ↓	63.90	44.61	41.72	41.64	57.49	41.76	40.65	44.19	40.22	41.43	40.10	**39.85**

Table 8: Effect of different top-*k* ratios used in latent matte propagation on VideoMatte.

Top-k ratio	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12
MSE ↓	7.28	7.26	7.25	**7.24**	7.36	7.50	7.64	7.82	8.01
MAD ↓	**40.30**	41.07	41.98	42.96	43.88	44.96	45.95	47.10	48.14

Latent Matte Perturbations. To get a robust matte decoder, we train it on both real image and video matting datasets. During training, we adopt different strategies to construct latent mattes for the two types of data. For video data, we encode video frames into VAE latents, add Gaussian noise according to the diffusion noise schedule at sampled timesteps, and feed the noisy latents into the DiT denoising process. The latent mattes are then obtained by applying the same LMP procedure as in inference. For image data, since temporal LMP cannot be applied to a single frame, we downsample the ground-truth alpha to the latent resolution and apply synthetic perturbations to approximate the artifacts observed in LMP-generated latent mattes. Compared with GT mattes, LMP latent mattes typically exhibit softer and slightly expanded boundaries, and may contain local foreground holes or sparse background activations.

Specifically, we simulate these artifacts with four perturbations: (1) *boundary expansion*, which blends the downsampled alpha with its 3×3 max-pooled version using a random interpolation weight; (2) *boundary softening*, which applies Gaussian blur with $\sigma \in [0.6, 1.4]$; (3) *foreground holes*, which randomly selects sparse foreground regions and attenuates their alpha values to mimic local missing responses; and (4) *background noise*, which injects random low-amplitude activations at sparse background locations. At each training iteration, the first two perturbations are always applied to model the systematic boundary characteristics of LMP latent mattes, while foreground holes and background noise are each triggered with a probability of 0.25. By randomly sampling the perturbation strengths within empirical ranges at each iteration, we expose the decoder to diverse imperfect latent mattes and improve its robustness during inference.

A.2 Additional Ablation Studies.

Attention Aggregation Range. We conduct ablation studies on the attention aggregation range used in latent matte propagation, including both the transformer layer range and the diffusion timestep range, and evaluate the propagated latent mattes on VideoMatte. As shown in Table 6, using only high layers leads to clearly worse performance, while aggregating a broader range of layers gives more reliable results. In particular, using all layers achieves the best MSE, suggesting that cross-layer aggregation provides more robust correspondences. Table 7 shows a similar trend for timestep selection. Early timesteps alone perform poorly due to heavy noise, whereas middle and late timesteps are more stable. Aggregating attention across all timesteps yields the best overall performance on both MSE and MAD. Based on these results, we use all layers and all timesteps in the final latent matte propagation setting.

Effect of Top-*k* Ratio. We further study the effect of the top-*k* ratio used in sparse affinity aggregation for latent matte propagation. As shown in Table 8, a relatively small top-*k* generally yields better results, while larger values gradually degrade both MSE and MAD due to the introduction of less reliable matches. Although the best MAD is achieved at smaller *k*, using a slightly larger ratio helps preserve more foreground responses during propagation, which is beneficial for maintaining object

Table 9: Effect of temperature hyper-parameter τ in affinity refinement on YouTubeMatte.

τ	MSE ↓	MAD ↓
1	**6.95**	**31.93**
2	11.57	45.38
3	21.28	66.82

Table 10: Sensitivity analysis of LMP post-processing thresholds t_{low} and t_{high} on YouTubeMatte (1920×1080), measured by overall MSE ↓ / boundary MSE ↓.

$t_{\text{low}} \setminus t_{\text{high}}$	0.70	0.80	0.90
0.10	12.67 / 70.69	12.15 / 66.73	12.16 / **64.95**
0.20	11.64 / 70.72	11.12 / 66.75	11.13 / 64.97
0.30	10.28 / 71.88	**9.76** / 67.91	9.78 / 66.13

Table 11: Comparison of different pseudo-label update strategies on YouTubeMatte (1920×1080).

Update Strategy	MAD ↓	MSE ↓	Grad ↓	dtSSD ↓
(a) Original S2D labels $\mathcal{O}$ only	2.19	0.83	7.99	1.65
(b) Direct replacement with predictions $\mathcal{N}$	1.78	0.54	6.04	1.48
(c) Selective update via SC-PT (Ours)	**1.54**	**0.47**	**5.54**	**1.42**

completeness. Therefore, we choose k=0.07 in the final setting. While this choice is not optimal in terms of MAD alone, it provides a better trade-off between error and foreground preservation, and the residual noise can be further reduced by the subsequent post-processing step.

Effect of Temperature τ. We evaluate the impact of the temperature parameter τ in Eq. (4) by varying its value on YouTubeMatte [31]. As shown in Table 9, $\tau = 1$ achieves the best performance in both MSE (6.95) and MAD (31.93), whereas larger values ($\tau = 2, 3$) substantially degrade propagation quality. This indicates that the unscaled RoPE-conditioned self-affinity already provides sufficient structural regularization, while further sharpening makes the propagation overly selective and prunes valid foreground correspondences. Consequently, we fix $\tau = 1$ without requiring additional hyperparameter tuning.

Post-Processing Thresholds. To evaluate the sensitivity of the two thresholds t_{low} and t_{high} in Eq. (7), we conduct a grid study on YouTubeMatte (1920×1080). We report both overall MSE and boundary MSE, which specifically evaluates fine-detail transition regions such as hair strands. As shown in Table 10, overall MSE varies smoothly within $[9.76, 12.67]$ across all combinations without abrupt degradation, and it is primarily influenced by t_{low}. Specifically, a moderately larger t_{low} suppresses more low-confidence background responses and reduces overall error. Meanwhile, boundary MSE remains stable within $[64.95, 71.88]$ and is mainly governed by t_{high}. Increasing t_{high} consistently improves boundary accuracy by preventing uncertain boundary activations from being over-enhanced. These results confirm that LMP post-processing is robust to threshold selection and preserves fine boundary structures.

Label Update Strategies in SC-PT. In Self-Corrected Progressive Training (SC-PT), we use the first-frame ground-truth alpha G_0 to evaluate the first-stage model predictions $\mathcal{N}$ against the original S2D labels $\mathcal{O}$, and selectively update the labels under temporal consistency constraints. To verify this design, we compare three strategies on YouTubeMatte (1920×1080) using MatAnyone [31]: (a) training both stages exclusively with the original S2D labels $\mathcal{O}$; (b) training the first stage with $\mathcal{O}$ and directly replacing all synthetic labels with first-stage predictions $\mathcal{N}$ in the second stage; and (c) our full SC-PT, which selectively combines $\mathcal{O}$ and $\mathcal{N}$ via first-frame quality gain and temporal consistency filtering. As shown in Table 11, both (a) and (b) achieve strong performance, demonstrating that both $\mathcal{O}$ and $\mathcal{N}$ provide effective supervision. SC-PT further improves all metrics and reduces MAD from 2.19 and 1.78 to 1.54, confirming the benefit of conservative label selection and temporal consistency filtering.

A.3 Further Analysis and Comparisons

Comparison with More Methods. We provide additional quantitative and efficiency comparisons with two more methods, MatAnyone 2 [30] and VideoMaMa [9], on YouTubeMatte. Specifically, MatAnyone 2 improves upon MatAnyone by training on VMReal, a large-scale real-world dataset with 28K video clips and 2.4M annotated frames. VideoMaMa formulates video matting as a video

Table 12: Quantitative comparison with more methods on YouTubeMatte.

Metric	MatAnyone [31]	MatAnyone 2 [30]	SAM2+VideoMaMa [9]	MatAnyone + S2D
YouTubeMatte (512×288*)*				
MAD ↓	2.72	**2.30**	–	2.33
MSE ↓	1.01	**0.78**	–	0.83
Grad ↓	0.97	0.78	–	**0.77**
dtSSD ↓	1.60	**1.45**	–	**1.45**
Conn ↓	0.39	**0.32**	–	**0.32**
YouTubeMatte (1920×1080*)*				
MAD ↓	1.99	1.61	**1.36**	1.54
MSE ↓	0.71	0.50	**0.29**	0.47
Grad ↓	8.91	7.13	**3.57**	5.54
dtSSD ↓	1.65	1.53	–	**1.42**

Table 13: Per-video inference efficiency comparison with more methods on YouTubeMatte.

Method	Inference Time (s) ↓	FPS ↑
MatAnyone + S2D (Ours)	**10.93**	**9.15**
MatAnyone 2 [30]	13.09	7.64
SAM2 + VideoMaMa [9]	56.51	1.77

Table 14: Adaptability of LMP across different video generative models on YouTubeMatte.

Video Generator	MSE ↓	MAD ↓	IoU ↑	Dice ↑
Wan2.2 [23]	**5.99**	**24.66**	**91.64**	**95.46**
Hunyuan-Video [7]	9.93	30.15	90.24	94.49
LTX-Video [3]	13.25	38.39	85.88	91.98

diffusion inference task to refine coarse segmentation masks into alpha mattes. Given a first-frame mask, the SAM2+VideoMaMa pipeline first runs SAM2 across the video and then performs diffusion denoising. In contrast, our MatAnyone+S2D keeps the original MatAnyone architecture and inference pipeline unchanged, and only updates the training dataset with S2D-VM under the SC-PT training protocol.

As shown in Table 12, MatAnyone+S2D achieves highly competitive results at 512×288 and outperforms MatAnyone 2 across all metrics at 1920×1080. In particular, it reduces MAD from 1.61 to 1.54, MSE from 0.50 to 0.47, Grad from 7.13 to 5.54, and dtSSD from 1.53 to 1.42. Furthermore, while SAM2+VideoMaMa attains lower pixel and gradient errors at 1920×1080, it incurs a much higher computational cost at test time. As reported in Table 13, SAM2+VideoMaMa requires 56.51 s per video (1.77 FPS), whereas MatAnyone+S2D takes only 10.93 s per video (9.15 FPS). Overall, our method runs approximately $5.2\times$ faster than SAM2+VideoMaMa and is also faster than MatAnyone 2 while maintaining strong high-resolution matting quality.

Generalization Across Video Generators. LMP propagates the first-frame alpha matte using cross-frame attention correspondences between the reference image and generated frames. It is not limited to Wan2.2 [23] and can be applied to other image-to-video diffusion transformers. To verify this adaptability, we apply LMP to two additional open-weight generators, including the 1.5-step distilled Hunyuan-Video [7] and the 2B LTX-Video [3]. We evaluate the propagated latent mattes using MSE and MAD for alpha accuracy, alongside IoU and Dice at a 0.5 threshold for coarse foreground localization. As shown in Table 14, without extensive per-model hyperparameter tuning, Hunyuan-Video and LTX-Video exhibit a gap in MSE and MAD compared with Wan2.2 due to differences in latent compression and attention distributions. Nevertheless, both models still achieve strong coarse localization performance. Specifically, Hunyuan-Video reaches 90.24 IoU and 94.49 Dice, closely approaching Wan2.2 at 91.64 IoU and 95.46 Dice, while LTX-Video also maintains 85.88 IoU and 91.98 Dice. These results demonstrate that LMP provides reliable coarse structural guidance across different video generative backbones.

Computational Cost. We evaluate the computational cost of the S2D data generation pipeline on a single NVIDIA RTX 4090 GPU. Table 15 reports the wall-clock time and peak GPU memory required to generate one 73-frame video at 1280×704 resolution, comparing video-only generation with full S2D generation. Most of the computational overhead comes from the 5B video generation backbone, which takes 299.44 s and 22905.6 MB of peak memory. Since LMP reuses the internal attention

Table 15: Wall-clock time and peak GPU memory for generating one 1280×704 video (73 frames) and its alpha matte on a single NVIDIA RTX 4090 GPU.

Pipeline Stage	Time per Video (s) ↓	Peak Memory (MB) ↓
Video generation only	299.44	22905.6
Video + alpha matte generation (Full S2D)	323.97	22958.2

representations from DiT denoising and the coarse-to-fine matte decoder is lightweight, generating the corresponding alpha mattes adds only 24.53 s in runtime and 52.6 MB in peak memory. Moreover, the S2D pipeline is executed offline for dataset construction and does not introduce additional overhead to downstream matting models during training or inference.

A.4 Limitations

Our method still has several limitations. First, S2D-VM is a synthetic dataset. Although the generated video-matte pairs are more realistic than conventional foreground-background composites, a gap to real-world videos still remains in some challenging cases, such as complex motion, scene dynamics, and appearance variations. Second, the S2D pipeline is relatively time-consuming, since dataset construction involves video generation and subsequent pseudo-label synthesis for each sample. This may make large-scale data expansion more computationally demanding than standard compositing-based pipelines.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction indeed reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The detailed experimental settings are included.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data and code will be publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The detailed experimental settings are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper includes extensive comparisons under different data scales, training data compositions, and training protocols, which demonstrate consistent empirical trends. However, we do not report repeated runs with error bars, confidence intervals, or formal statistical significance tests, and therefore answer No.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources are reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research strictly follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work provides novel perspectives for video matting data generation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release a general-purpose image or video generator, nor does it rely on a scraped Internet-scale dataset with clear high-risk misuse potential. Our contribution focuses on a synthetic data generation paradigm for video matting and a task-specific dataset derived from existing matting sources, so this question is not directly applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original papers of the datasets used are all properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: There are no new assets in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: There are no crowdsourcing experiments or research with human subjects in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not include study participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Although an off-the-shelf vision-language model is used to generate descriptive prompts during dataset construction, LLM/VLM usage is not an important, original, or non-standard component of the core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.