Wan-Alpha++: Animating a Single RGBA Asset into Controllable Transparent Video Effects

Haotian Dong, Xin Wang, Wenjing Wang, *Member, IEEE*, Chen Li, Jing Lyu, Yan Tina Luximon, Ping Li, *Member, IEEE*, Wanchao Su, Ke Xu, Di Lin, *Member, IEEE*

arXiv:2610.26189v1 [cs.CV] 19 Oct 2026

Abstract—RGBA video generation is essential for creating compositable visual assets in filmmaking, game development, and digital content production. However, generating high-quality RGB appearance alongside temporally coherent transparency remains a significant challenge. Existing text-to-RGBA video methods often entangle RGB and alpha distributions and offer limited control over the precise appearance of a target asset. In addition, incorporating image conditioning typically suffers from the emergence-disappearance effects, which fail existing first-frame or first-and-last-frame conditions due to their fully transparent boundary frames. To address these issues, we present Wan-Alpha++, a unified framework that brings controllable RGBA video generation to both text- and image-to-video diffusion backbones through a shared RGBA latent space, extending our conference version Wan-Alpha from the text-conditioned setting to image-conditioned generation with an interior reference frame. Our method first learns shiftable RGB-alpha distributions while preserving the pretrained RGB prior. Specifically, we introduce a transparency-aware bidirectional diffusion objective to produce distinguishable RGBA latent representations, and a transparency-guided mean sampler that shifts the noise distribution to improve transparency generation and enable spatial controllability. Building upon this RGBA generation prior, we introduce temporal phase reparameterization for image-conditioned effect generation. By cyclically transforming a zero-positive-zero alpha pattern into a positive-zero-positive pattern, it relocates visible content to both temporal boundaries, facilitating effective appearance conditioning with limited RGBA data. During inference, normalized transparent-state gradient guidance places a user-specified invisible interval, and an inverse phase transformation subsequently recovers the desired emergence-disappearance effect. Extensive experiments demonstrate that Wan-Alpha++ achieves state-of-the-art performance in text-to-RGBA generation, precise spatial transparency control, and inference efficiency. For image-to-RGBA effect generation, Wan-Alpha++ significantly outperforms previous generation-and-matting pipelines, improving DINO-I from 0.6679 to 0.9227 and CLIP-I from 0.7969 to 0.9368, while providing enhanced temporal consistency and controllable transparency evolution.

Index Terms—RGBA video generation, Image-to-video, Temporal control

1 Introduction

RGBA extends conventional RGB representation with an alpha channel that describes per-pixel transparency, enabling foreground content to be directly composited with arbitrary backgrounds. As a result, RGBA videos are widely used in game development, filmmaking, television production, user-interface design, and other creative workflows. While automatically generating RGBA videos can substantially reduce manual effort, it faces unique challenges beyond standard RGB video generation: the model must jointly synthesize high-quality RGB appearance, spatially accurate transparency, and temporally coherent alpha evolution.

Early text-to-RGBA video generation methods [1], [2], [3], [4] extend RGBA image generation models to video frameworks. A representative strategy combines the 2D RGBA VAE of LayerDiffuse [5] with pretrained video diffusion models such as AnimateDiff [6]. However, an image VAE does not explicitly model temporal consistency and tends to entangle RGB and alpha information in the latent space, resulting in degraded appearance, inaccurate transparency, and limited motion. Subsequent approaches improve temporal modeling through dedicated video architectures. TransPixeler [7], for example, introduces alpha tokens, duplicates the generation backbone, and employs cross-RGBA attention for information exchange. Nevertheless, architectural interaction alone does not fully resolve the ambiguity between RGB appearance and alpha transparency; it also incurs substantial computational overhead and remains limited in

- *Haotian Dong is with the College of Intelligence and Computing, Tianjin University, China.*
- *Xin Wang is with the Laboratory for Artificial Intelligence in Design Limited (AiDLab), The Hong Kong Polytechnic University, Hong Kong.*
- *Wenjing Wang is an independent researcher.*
- *Chen Li is an independent researcher.*
- *Jing Lyu is an independent researcher.*
- *Yan Luximon is with the School of Design, The Hong Kong Polytechnic University, Hong Kong.*
- *Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: lipingfire@gmail.com).*
- *Wanchao Su is with the School of Software, Dalian University of Technology, China and SensiLab, Monash University, Australia.*
- *Ke Xu is with the University of Science and Technology of China, China*
- *Di Lin is with the College of Intelligence and Computing, Tianjin University, China.*

synthesizing semi-transparent effects such as smoke, illumination, water, and translucent materials.

To address these text-driven challenges, our preliminary conference work, Wan-Alpha [8], explores RGB-alpha distribution learning for text-to-RGBA video generation. Instead of substantially modifying the pretrained video-generation backbone, it preserves the learned RGB distribution while shifting the alpha distribution in both latent and noise spaces. Through a transparency-aware bidirectional diffusion objective that produces more distinguishable RGBA latent representations, a transparency-guided mean sampler that enhances transparency generation and enables spatial control, and a carefully constructed RGBA video dataset, Wan-Alpha can generate stable transparency without compromising the RGB generation capability of the pretrained model.

Nonetheless, text-to-RGBA video generation methods still fall short in achieving precise control over the visual appearance and spatial composition. An RGBA reference image offers explicit appearance and transparency information, yet directly extending existing image-to-RGB models to RGBA video generation remains a challenging and largely unexplored task. A fundamental limitation arises from their first-frame or first-and-last-frame conditioning mechanisms [9], [10], [11], [12], [13], which assume that the target content is observable at temporal boundaries of the generated video. This assumption breaks down for RGBA effects, where complete disappearance of content at the first and last frames is a common and often desirable design choice. A straightforward solution is to directly modify the condition mask to support an interior reference, which requires a large amount of data that is prohibitively expensive to collect. Alternatively, inference-time gradient guidance may optimize a designated interior frame toward the reference image, but such a soft constraint cannot guarantee exact appearance and alpha preservation and may compromise generation quality. Consequently, middle-frame-conditioned RGBA video generation faces two coupled challenges: modality adaptation from RGB to RGBA generation and temporal-conditioning adaptation from boundary to interior reference frames.

In this paper, we extend our Wan-Alpha [8] to address the above two challenges, by proposing Wan-Alpha++, a unified framework for controllable text-to-RGBA and data-efficient image-to-RGBA video generation. Instead of directly modifying the conditioning pattern of the pretrained first-and-last-frame video model, we reparameterize the training distribution through visibility-aware temporal rotation. Specifically, an RGBA effect that emerges from and subsequently vanishes into transparency is cyclically shifted such that its content-rich interior frames are moved to the temporal boundaries. The resulting sequences contain observable content at both endpoints, allowing us to retain the boundary-conditioning capability of the pretrained model while learning the distinctive temporal evolution of RGBA effects from limited data. Based on the learned distribution, we introduce transparent-state guidance during inference: the same reference image is supplied at both boundaries, while a user-specified interior interval is guided toward a fully transparent target by minimizing a pixel-space MSE loss along the sampling trajectory. Unlike an arbitrary RGB target, a fully transparent frame represents a low-complexity state and can therefore be reliably imposed through gradient guidance. Finally, we apply the inverse temporal rotation to recover the desired emergence-disappearance sequence. These two designs are mutually dependent: temporal rotation converts difficult interior conditioning into familiar boundary conditioning, while transparent-state guidance determines the invisible interval required to recover the target temporal pattern.

We summarize our major contributions below:

- We propose Wan-Alpha, a high-quality text-to-RGBA video generation framework based on shiftable RGB–alpha distribution learning. It enables stable and controllable transparency generation while preserving the RGB synthesis capability of a pretrained video model.
- We introduce a transparency-aware bidirectional diffusion objective for distinguishable RGBA latent learning and a transparency-guided mean-shifting strategy for controllable RGBA generation. These designs improve joint RGB-alpha modeling in the latent and noise spaces, respectively.
- We construct a high-quality RGBA video dataset containing diverse motion, complex boundaries, and semi-transparent effects. Together with the proposed framework, it enables state-of-the-art performance across generation quality, temporal consistency, transparency rendering, and controllability.
- We propose Wan-Alpha++, which extends Wan-Alpha from text conditioning to image conditioning. By introducing temporal phase reparameterization to convert an interior reference into observable boundary conditions, and transparent-state gradient guidance to control the invisible interval and recover the desired emergence-disappearance effect, Wan-Alpha++ achieves middle-frame-conditioned RGBA video generation under limited training data.

2 Related Work

2.1 RGBA Image Generation

Compared with RGB images [14], [15], [16], [17], [18], [19], [20], [21], high-quality RGBA images are

substantially more difficult to acquire, making it expensive to train large-scale RGBA generative models from scratch. Existing approaches therefore commonly adapt pretrained RGB image models to transparency generation. LayerDiffuse [5] introduces latent transparency by adapting the VAE of Stable Diffusion [22] to encode RGB and alpha information within a shared latent space, followed by parameter-efficient diffusion fine-tuning. AlphaVAE [23] improves the RGBA representation and substantially reduces the amount of training data required for VAE adaptation. In contrast, Alfie [24] extracts transparency from the attention features of a pretrained diffusion model without additional training, but mainly supports transparent backgrounds and has limited ability to represent semi-transparent content. Subsequent studies extend RGBA generation to multi-layer image synthesis [25], [26], [27], [28] and transparent image inpainting [29].

Although these methods demonstrate the potential of pretrained RGB priors for transparency generation, they operate on individual images and do not explicitly model temporal alpha evolution.

2.2 RGBA Video Generation

RGBA video generation additionally requires temporally coherent motion and transparency, while suitable training data remain scarce. Early approaches [1], [2], [3], [4] combine the image-based LayerDiffuse VAE [5] with video generation frameworks such as AnimateDiff [6], but the lack of explicit temporal modeling in the image VAE often results in unstable alpha channels and limited motion. Other studies develop dedicated architectures for transparent or layered video generation. LayerFlow [30] formulates layer-aware video synthesis within a unified generation framework, while TransPixeler [7] duplicates the video backbone and introduces cross-RGBA attention to exchange information between RGB and alpha branches.

However, these methods either incur substantial computational overhead or remain limited in distinguishing the distribution of RGB appearance from that of transparency. Our preliminary conference work Wan-Alpha [8] learns shiftable RGB–alpha distributions in the latent and noise spaces, enabling stable and controllable text-to-RGBA video generation while preserving the RGB synthesis capability of the pretrained model. Despite being able to generate high-quality and diverse videos, text-to-RGBA video generators typically offer less satisfactory spatial controllability (*e.g.*, animating the appearance and alpha structure of a user-provided asset).

2.3 Image-to-RGB Video Generation

We review related Image-to-RGB video generation methods, as Wan-Alpha++ is the first Image-to-RGBA video generation model.

Boundary-Frame-Conditioned Generation Recent image-to-video generation (I2V) methods generally use reference image(s) to constrain visual appearance while synthesizing temporal dynamics. A common paradigm places the reference at the first frame. Stable Video Diffusion [31] adapts latent video diffusion models to animate a given image. DynamiCrafter [32] incorporates both latent-level image features and pixel-level image information into a pretrained text-to-video model. CogVideoX-I2V [11] encodes the input image with a 3D VAE, pads its latent representation along the temporal dimension, and concatenates it with the noisy video latent. Wan [9] further scales this paradigm to large diffusion transformers. FrameBridge [33] reformulates I2V as a data-to-data bridge process, treating the input image as the starting point of generation instead of an auxiliary condition.

Another line of work incorporates both first- and last-frame conditions to constrain video endpoints. PixelDance [34] uses a strict first-frame condition together with an optional last-frame instruction to generate motion-rich videos. SEINE [35] adopts random-mask training and supports first-and-last-frame transition as one of several frame-completion configurations. ToonCrafter [36] specializes in dual-frame conditioning for generative cartoon interpolation, while ViBiDSampler [37] performs training-free bidirectional sampling by alternately conditioning forward and backward generation on the two endpoints. Wan2.1-FLF2V [9] directly trains a large video model with first-and-last-frame conditions.

These methods demonstrate the effectiveness of boundary images for appearance and endpoint control. However, their conditioning mechanisms are tied to predefined temporal positions and are unsuitable when the informative reference corresponds to an interior frame, as frequently occurs in emergence–disappearance RGBA effects.

Training-Free Guidance for Video Generation Training-free diffusion guidance provides an alternative by controlling pretrained models without fine-tuning on specific tasks. FreeDoM [38] and MPGD [39] construct differentiable objectives to guide diffusion sampling while preserving the pretrained generative prior. Recent studies extend this paradigm to video generation. Specifically, MOFT [40] extracts motion-aware features from video diffusion models and optimizes noisy latents using feature-correlation losses for training-free motion control. TITAN-Guide [41] further introduces memory-efficient forward-gradient optimization for

text-to-video diffusion models and demonstrates its applicability to frame interpolation. FlexTI2V [42] supports visual conditions at arbitrary temporal positions through dynamic latent-patch exchange, although it does not explicitly use gradient guidance.

These approaches improve conditioning flexibility without model retraining, but generally impose soft sampling-time constraints and cannot guarantee exact reproduction of a target frame.

3 METHOD

3.1 Overview

Wan-Alpha++ aims to generate an RGBA video from a user-provided RGBA image while faithfully preserving its appearance and enabling controllable emergence and disappearance process. Building upon our preliminary text-to-RGBA video generation framework [8], Wan-Alpha++ substantially advances the previous version and learns distinguishable RGB-alpha representations and controllable transparency distributions. We first revisit the RGBA VAE training architecture in Sec. 3.2, separating RGB and alpha information in a latent space. Based on this RGBA representation prior, we construct an image-conditioned model in Sec. 3.3. We first learn a controllable spatial distribution by introducing a transparency-guided sampler. Directly learning middle-frame conditioning requires adapting the pretrained image-to-video model to a temporal conditioning pattern unseen during pretraining, which typically demands substantial training data. This is particularly costly for RGBA video generation because high-quality RGBA videos are scarce and expensive to collect. We instead reparameterize the temporal phase of RGBA effects, converting a sequence whose informative content lies in the interior into one with observable boundary frames. This allows the model to retain the pretrained first-and-last-frame conditioning mechanism. During inference, transparent-state guidance determines an interior invisible interval, after which an inverse temporal transformation recovers the desired emergence-disappearance effect. In Sec. 3.4, we introduce the construction of our RGBA dataset for VAE and DiT. All components enable our Wan-Alpha++ to achieve state-of-the-art performance across multiple metrics.

3.2 Shiftable RGBA Distribution Learning

3.2.1 Distinguishable RGBA Latent Representation

As illustrated in Figure 1, we start from a pretrained RGB video generation model [9] consisting of a video VAE and a diffusion transformer. Given an RGBA video, we denote the video V as a combination of RGB component V_{rgb} and transparency component V_α, i.e., $V = (V_{rgb}, V_\alpha)$. We aim to embed RGB appearance and alpha transparency into a shared latent representation without changing its original dimensionality. This enables the pretrained video diffusion model to be adapted for RGBA generation without requiring any modifications to its backbone architecture.

Directly encoding V_{rgb} and V_α into the same latent space tends to entangle background color with transparency. We therefore introduce a hard and a soft rendering function:

$$\begin{cases} R_h(V_{\text{rgb}}, V_\alpha, c) = V_{\text{rgb}} \odot \mathbb{I}(V_\alpha > 0) + c \odot [1 - \mathbb{I}(V_\alpha > 0)], \\ R_s(V_{\text{rgb}}, V_\alpha, c) = V_{\text{rgb}} \odot V_\alpha + c \odot (1 - V_\alpha), \end{cases} \tag{1}$$

where $\odot$ denotes element-wise multiplication, $\mathbb{I}(\cdot)$ is an indicator function, and c is a randomly sampled color value from a predefined color set C={black, blue, green, cyan, red, magenta, yellow, white}.

The hard-rendered RGB video and the three-channel replicated alpha video are separately encoded by a frozen video encoder E. A trainable feature merge block $\mathcal{M}$ then produces the shared RGBA latent:

$$Z = \mathcal{M}\left(E\left(R_h(V_{\text{rgb}}, V_\alpha, c)\right), E(V_\alpha)\right). \tag{2}$$

The latent Z is decoded by two branches equipped with RGB and alpha LoRA parameters: $\widehat{V}_{\text{rgb}} = D_{\text{rgb}}(Z)$ and $\widehat{V}_\alpha = D_\alpha(Z)$.

Given a reconstruction $\widehat{X}$ and its target X, we formulate pixel, perceptual, and edge-consistency objectives to supervise reconstruction under different transparency states as:

$$\begin{cases} \mathcal{L}_{\text{pix}}(\widehat{X}, X) = \|\widehat{X} - X\|_1, \\ \mathcal{L}_{\text{perc}}(\widehat{X}, X) = \|\Phi(\widehat{X}) - \Phi(X)\|_2^2, \\ \mathcal{L}_{\text{edge}}(\widehat{X}, X) = \|S(\widehat{X}) - S(X)\|_1, \\ \mathcal{L}_{\text{rec}}(\widehat{X}, X) = \mathcal{L}_{\text{pix}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}}, \end{cases} \tag{3}$$

where Φ is a pretrained perceptual feature extractor, S denotes the Sobel operator, and $\mathcal{L}_{\text{rec}}$ is the overall reconstruction objective. Using independently sampled colors c_s and c_h, the reconstructed rendered videos $\widehat{V}^s_{\text{rgb}}$ and $\widehat{V}^h_{\text{rgb}}$, and the corresponding ground-truth rendered videos V^s_{rgb} and V^h_{rgb} are formulated as:

$$\begin{cases} \widehat{V}^s_{\text{rgb}} = R_s(\widehat{V}_{\text{rgb}}, \widehat{V}_\alpha, c_s), \ V^s_{\text{rgb}} = R_s(V_{\text{rgb}}, V_\alpha, c_s), \\ \widehat{V}^h_{\text{rgb}} = R_h(\widehat{V}_{\text{rgb}}, \widehat{V}_\alpha, c_h), \ V^h_{\text{rgb}} = R_h(V_{\text{rgb}}, V_\alpha, c_h). \end{cases} \tag{4}$$

The rendering objective $\mathcal{L}_{\text{render}}$ is then formulated as:

$$\begin{cases} \mathcal{L}_\alpha = \mathcal{L}_{\text{rec}}(\widehat{V}_\alpha, V_\alpha), \ \mathcal{L}^s_{\text{rgb}} = \mathcal{L}_{\text{rec}}(\widehat{V}^s_{\text{rgb}}, V^s_{\text{rgb}}), \\ \mathcal{L}^h_{\text{rgb}} = \mathcal{L}_{\text{rec}}(\widehat{V}^h_{\text{rgb}}, V^h_{\text{rgb}}), \mathcal{L}_{\text{render}} = \mathcal{L}_\alpha + \mathcal{L}^s_{\text{rgb}} + \mathcal{L}^h_{\text{rgb}}. \end{cases} \tag{5}$$

Transparency-Aware Bidirectional Diffusion Objective Preserving the original RGB latent distribution maintains the generative capability of the pretrained DiT, whereas shifting alpha-related latents makes transparency more distinguishable. Measuring only

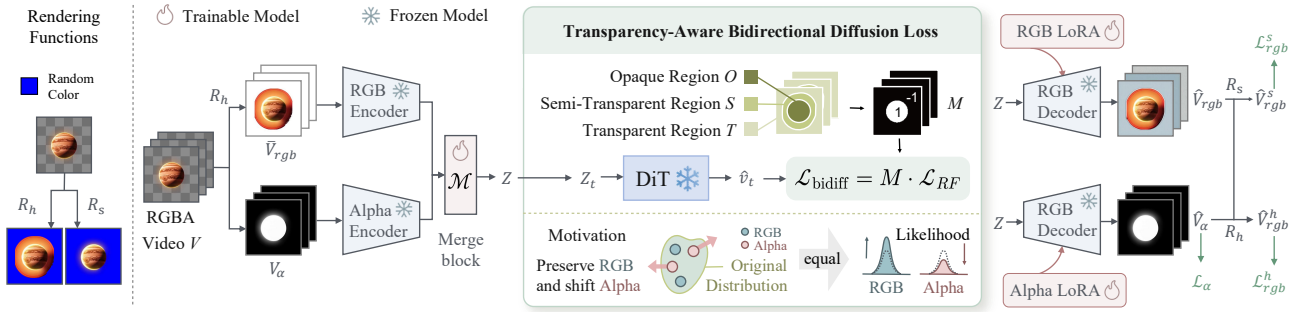


Fig. 1. Training framework of our RGBA VAE. The RGB and alpha channels are encoded separately and fused into a shared latent representation. The transparency-aware bidirectional diffusion objective preserves the pretrained RGB distribution while shifting alpha-related representations. Rendering-based reconstruction objectives further supervise RGB appearance and alpha transparency.

statistical distances between latent features, however, does not guarantee that these representations are distinguishable to the video DiT. We therefore introduce the frozen DiT into VAE training and optimize the latent representation from its perspective.

Let $t \in [0,1]$ denote the rectified-flow timestep. The noisy latent and target velocity are:

$$Z_t = (1-t)Z + t\epsilon, \quad v_t = \epsilon - Z, \quad \epsilon \sim \mathcal{N}(0, I). \tag{6}$$

Given the velocity $\widehat{v}_t$ predicted by the frozen DiT, the point-wise rectified-flow loss is:

$$\ell_{\mathrm{RF}}(p) = \|\widehat{v}_t(p) - v_t(p)\|_2^2, \tag{7}$$

where p indexes a spatial-temporal location in the latent space. Aggregating the point-wise errors over all spatial-temporal latent locations gives the standard rectified-flow objective:

$$\mathcal{L}_{\mathrm{RF}} = \frac{1}{|\Omega|}\sum_{p\in\Omega}\ell_{\mathrm{RF}}(p) = \frac{1}{|\Omega|}\sum_{p\in\Omega}\|\widehat{v}_t(p) - v_t(p)\|_2^2, \tag{8}$$

where Ω denotes the set of all spatial-temporal locations in the latent representation. This standard objective assigns equal importance to opaque, semi-transparent, and fully transparent regions.

We resize the alpha video to the latent resolution and define a signed mask:

$$M(p) = \begin{cases} 1, & p \in \mathcal{O}, \\ -1, & p \in \mathcal{S} \cup \mathcal{T}, \end{cases} \tag{9}$$

where $\mathcal{O}$, $\mathcal{S}$, and $\mathcal{T}$ denote opaque, semi-transparent, and fully transparent regions, respectively. The transparency-aware bidirectional diffusion objective is:

$$\mathcal{L}_{\mathrm{bidiff}} = M \cdot \mathcal{L}_{RF}. \tag{10}$$

Positive gradients preserve high likelihood for opaque RGB content, whereas reversed gradients shift non-opaque representations away from the original RGB distribution. The complete VAE objective is:

$$\mathcal{L}_{\mathrm{VAE}} = \mathcal{L}_{\mathrm{render}} + \mathcal{L}_{\mathrm{bidiff}}. \tag{11}$$

3.3 Controllable Image-to-RGBA Generation

In Figure 2, we introduce the training framework of our RGBA video generator. Figure 3 further illustrates the transparency-guided noise distribution. Based on the trained DiT, Figure 4 presents the process of transparent-state gradient guidance during inference.

3.3.1 *Boundary-Conditioned RGBA Initialization*

We first transfer the learned RGBA representation to a pretrained image-to-video model. Given a reference image, its encoded latent is placed at the first temporal position, while the remaining positions are zero-padded. A binary temporal mask indicates the position of the image condition. We initially train on generic RGBA videos to obtain an image-to-RGBA video generation model and subsequently extend the same conditioning interface to the first-and-last-frame setting.

Given an RGBA video $V = \{V_i\}_{i=0}^{T-1}$, the first-and-last-frame condition is represented as:

$$C_i = \begin{cases} E_{\mathrm{RGBA}}(V_0), & i = 0, \\ E_{\mathrm{RGBA}}(V_{T-1}), & i = T-1, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

with the corresponding temporal mask:

$$B_i = \mathbb{I}\left(i \in \{0, T-1\}\right). \tag{13}$$

The condition latent C, condition mask B, and noisy video latent are jointly provided to the denoising model. This initialization transfers the appearance-conditioning capability of the RGB video foundation model to the RGBA domain, but it still assumes that informative content exists at the temporal boundaries.

3.3.2 *Visibility-Aware Temporal Phase Reparameterization*

Many RGBA effects follow an emergence-disappearance pattern. As illustrated in Figure 2, they start from a fully transparent state, gradually

become visible, and eventually fade out. For frame index i, we define a temporal alpha mass m as:

$$m_i = \frac{1}{HW} \sum_{x=1}^{W} \sum_{y=1}^{H} V_{\alpha,i}(x,y). \tag{14}$$

An emergence-disappearance effect approximately satisfies:

$$m_0 \approx 0, \qquad \max_i m_i > 0, \qquad m_{T-1} \approx 0. \tag{15}$$

Its original zero-positive-zero pattern therefore contains no effective appearance information at either temporal boundary.

Instead of moving the image condition to an interior position and retraining the conditioning interface, we move the temporal origin of the video. We design a cyclic temporal transformation $\mathcal{R}_k$ to reparameterizes V into a phase-shifted sequence $\widetilde{V}$ by setting its k-th frame as the new temporal origin:

$$\widetilde{V} = \mathcal{R}_k(V), \;\; \widetilde{V}_i = V_{(i+k) \bmod T}, \;\; i = 0, \ldots, T-1, \tag{16}$$

To ensure that both transformed boundaries contain observable content, the cutting position k is sampled from:

$$k \sim \mathcal{K}, \;\; \mathcal{K} = \left\{ j \;\middle|\; m_j > \tau_m, \; m_{(j-1) \bmod T} > \tau_m \right\}, \tag{17}$$

where τ_m is a visibility threshold. Consequently, the existence pattern becomes:

$$0 \to + \to 0 \quad \xrightarrow{\mathcal{R}_k} \quad + \to 0 \to +. \tag{18}$$

The transformation changes only the temporal phase and preserves the frames, motion, and transparency evolution of the original effect. We use $\widetilde{V}_0$ and $\widetilde{V}_{T-1}$ as image conditions and fine-tune the RGBA FLF2V model on $\widetilde{V}$. Sampling k from $\mathcal{K}$ introduces phase diversity while maintaining valid boundary conditions. This strategy converts the challenging middle-frame conditioning problem into a simpler boundary-conditioning pattern already supported by the pre-trained model.

3.3.3 Transparency-Guided Noise Distribution

As illustrated in Figure 2, after learning the shared RGBA representation, we adapt the DiT to RGBA video generation. In Figure 3, we further introduce the transparency-guided noise distribution in detail. Standard rectified flow samples noise from a zero-mean Gaussian distribution and therefore provides no prior regarding the spatial arrangement of transparency. We instead shift the noise mean using a spatial mask derived from the alpha channel: $\widetilde{\epsilon} \sim \mathcal{N}(G \cdot \mu, I)$, where $G \in [0,1]^{T' \times H' \times W'}$ is a Gaussian ellipse mask and μ controls the mean-shift strength.

To construct G, we resize and binarize each alpha frame A_i and collect its positive-support coordinates $\mathcal{P}_i$, which is formulated as:

$$\begin{cases} B_i(x,y) = \mathbb{I}\left(A_i(x,y) > \tau_\alpha\right), \\ \mathcal{P}_i = \{(x,y) \mid B_i(x,y) = 1\}, \end{cases} \tag{19}$$

where τ_α is set as 0.5. The coordinate mean $\boldsymbol{m}$ and covariance Σ are formulated as:

$$\begin{cases} \boldsymbol{m}_i = \dfrac{1}{|\mathcal{P}_i|} \displaystyle\sum_{\boldsymbol{q} \in \mathcal{P}_i} \boldsymbol{q}, \\ \Sigma_i = \dfrac{1}{|\mathcal{P}_i| - 1} \displaystyle\sum_{\boldsymbol{q} \in \mathcal{P}_i} (\boldsymbol{q} - \boldsymbol{m}_i)(\boldsymbol{q} - \boldsymbol{m}_i)^\top. \end{cases} \tag{20}$$

Eigenvalue decomposition $\Sigma_i = U_i \Lambda_i U_i^\top$ provides the principal directions and orientation of the fitted ellipse. For coordinates (x_i', y_i') expressed in the principal-axis coordinate system, the Gaussian ellipse G is formulated as: $G_i(x,y) = \exp\left[-\frac{1}{2}\left(\frac{x_i'^2}{(a_i/2)^2} + \frac{y_i'^2}{(b_i/2)^2}\right)\right]$, where a_i and b_i are the lengths of the major and minor axes.

The shifted rectified-flow path and target velocity are:

$$Z_n = (1-n)Z + n\widetilde{\epsilon}, \qquad \widetilde{v}_n = \widetilde{\epsilon} - Z. \tag{21}$$

The DiT is optimized with $\mathcal{L}_{\mathrm{DiT}} = \|\widehat{v}_n - \widetilde{v}_n\|_2^2$. During inference, G can be specified by the user to control the approximate location and extent of generated content.

3.3.4 Transparent-State Gradient Guidance

After phase reparameterization, the model learns the positive-zero-positive temporal pattern. As illustrated in Figure 4, during inference, we place the same user-provided RGBA image I_{ref} at both temporal boundaries C_0 and C_{T-1}, which is formulated as:

$$C_0 = C_{T-1} = E_{\mathrm{RGBA}}(I_{\mathrm{ref}}). \tag{22}$$

These duplicated conditions anchor the appearance and spatial alpha structure of the generated effect. The user further specifies an interior interval $\mathcal{I}_{\mathrm{tr}}$ as:

$$\mathcal{I}_{\mathrm{tr}} = \{i_s, i_s + 1, \ldots, i_e\}, \tag{23}$$

where s and e denote the indices of the start and end frames, respectively. in which the output should become fully transparent. We set this as the middle eight frames by default.

At rectified-flow timestep n, the clean latent $\widehat{Z}_0$ is estimated from the current latent Z_n and predicted velocity $\widehat{v}_n$, which is formulated as:

$$\widehat{Z}_0 = Z_n - n\widehat{v}_n. \tag{24}$$

We decode the frames associated with $\mathcal{I}_{\mathrm{tr}}$:

$$\widehat{V}_{\mathcal{I}_{\mathrm{tr}}} = D_{\mathrm{RGBA}}\left(\widehat{Z}_0^{\mathcal{I}_{\mathrm{tr}}}\right). \tag{25}$$

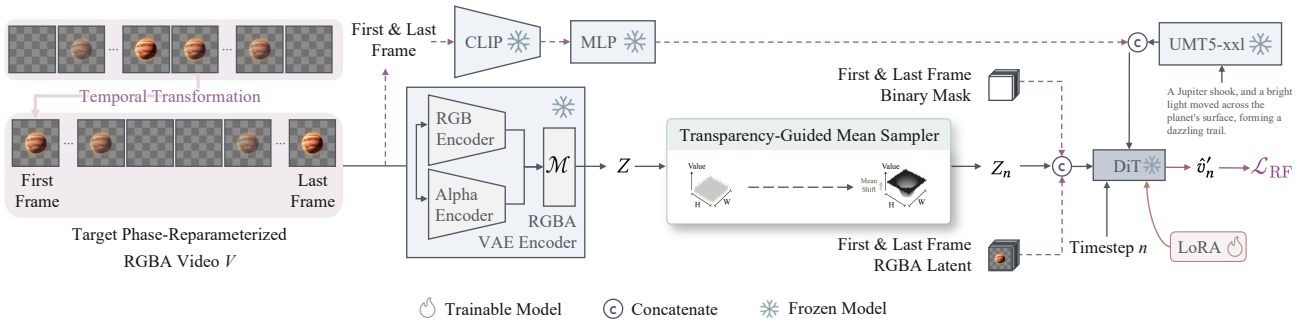


Fig. 2. Training framework of our RGBA video generator. The pretrained RGBA VAE encodes the input video into the shared latent space. Transparency-guided mean sampler provides a transparency prior for DiT adaptation.

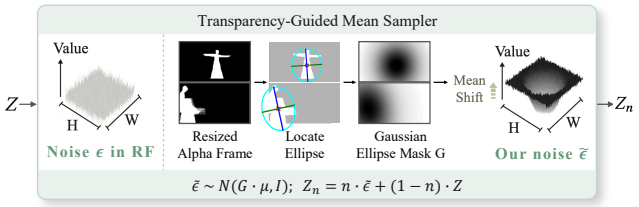


Fig. 3. A Gaussian ellipse derived from the alpha channel shifts the noise mean, providing a transparency prior for DiT adaptation. During inference, the mask can be specified by users to control the approximate position and extent of generated content.

Let $V_\varnothing$ denote a fully transparent RGBA target. The transparent-state objective $\mathcal{L}_{\text{tr}}$ is formulated as:

$$\mathcal{L}_{\text{tr}} = \frac{1}{|\mathcal{I}_{\text{tr}}|} \sum_{i \in \mathcal{I}_{\text{tr}}} \left\| \widehat{V}_i - V_\varnothing \right\|_2^2. \tag{26}$$

The current video latent is updated through inference-time gradient guidance. Let $\mathcal{G}$ denote the indices of the guided latent frames. We compute the gradient only with respect to the corresponding clean latent:

$$g_n = \nabla_{\widehat{Z}_{0,\mathcal{G}}} \mathcal{L}_{\text{tr}}, \bar{g}_n = \frac{g_n}{\|g_n\|_2 + \epsilon}, \widehat{Z}'_{0,\mathcal{G}} = \widehat{Z}_{0,\mathcal{G}} - \eta_n \bar{g}_n. \tag{27}$$

where g_n and $\bar{g}_n$ denote the raw and normalized guidance gradients, respectively. The parameter η_n controls the guidance step size, while ϵ is a small constant for numerical stability. The remaining latent frames are kept unchanged. The updated clean latent frames are subsequently re-noised to timestep n and merged with the unchanged frames, yielding the guided noisy latent Z'_n for the next denoising operation.

An arbitrary RGB keyframe contains complex color, texture, geometry, and spatial layout, making exact inference-time matching difficult. In contrast, a fully transparent frame represents a low-complexity state dominated by zero alpha occupancy. It can therefore be enforced more reliably through gradient guidance, while the complex appearance of the effect remains controlled by the trained boundary-conditioning mechanism.

3.3.5 *Inverse Phase Transformation*

The guided model produces a phase-reparameterized video following the positive-zero-positive pattern. We finally apply the inverse cyclic transformation:

$$V^* = \mathcal{R}_k^{-1}(\widetilde{V}). \tag{28}$$

This inverse transformation converts the generated sequence back to the target zero-positive-zero pattern:

$$+ \to 0 \to + \quad \xrightarrow{\mathcal{R}_k^{-1}} \quad 0 \to + \to 0. \tag{29}$$

Temporal phase reparameterization and transparent-state guidance are mutually dependent. The former converts an interior appearance condition into observable boundary conditions that can be learned from limited RGBA data, whereas the latter determines the zero-alpha interval required to recover the original effect after inverse transformation.

3.4 Dataset

High-quality RGBA videos remain scarce because their alpha channels typically require professional content creation or accurate video matting. We therefore construct the training data from multiple public image and video matting datasets, supplemented with RGBA visual effects collected from the Internet. Samples can be found in Figure 5.

RGBA VAE Data For VAE training, we collect images from ten image matting datasets, including AIM-500 [43], AM-2K [44], Distinctions-646 [45], HHM2K [46], HHM50K [46], Human2K [47], P3M-10K [48], RealWorldPortrait-636 [49], SIMD [50], and Transparent-460 [51]. Video samples are obtained from DVM [52], VideoMatting108 [53], VideoMatte240K [54], and additional Internet sources. Image samples are converted into static video clips and processed using randomized temporal windows so that they can be jointly trained with video data. We divide the resulting collection into 95% training and

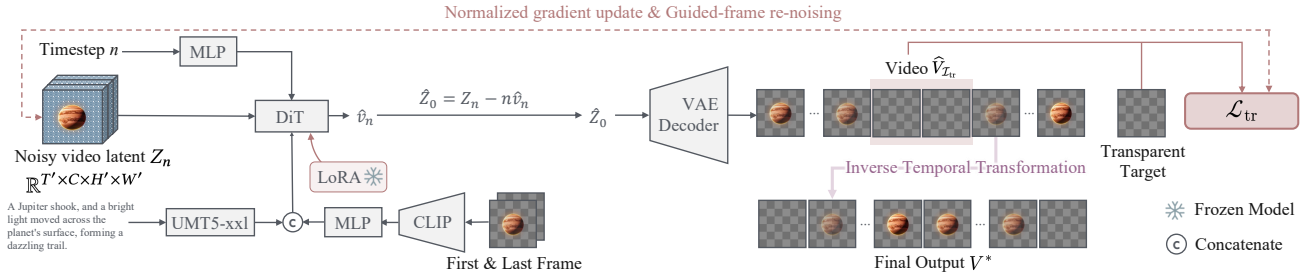


Fig. 4. Inference framework of the proposed phase-reparameterized image-to-RGBA video generation. The same input RGBA image conditions both temporal boundaries. Transparent-state guidance applies normalized gradient updates to a user-specified invisible interval. Finally, temporal transformation recovers the desired emergence-disappearance effect.

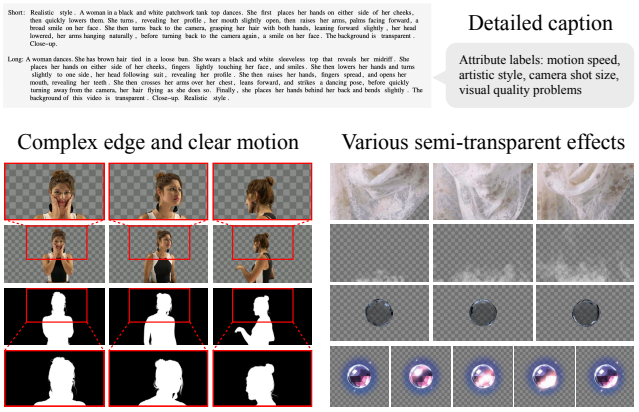


Fig. 5. Samples from our dataset. Our videos show clear motion and complex edges, such as hair. Each video has short and long captions, plus attribute labels. The dataset also includes various semi-transparent effects like sheer fabric, smoke, water, and glow.

5% validation subsets, yielding 77,237 training clips and 4,066 validation clips.

Text-to-RGBA Generation Data The diffusion transformer is adapted using a carefully selected subset of the same collection. We prioritize samples with clear motion, complex alpha boundaries, semi-transparent content, and lighting effects. For each sample, Qwen2.5-VL-72B [55] is used to generate both short and detailed captions. We additionally annotate motion speed, artistic style, camera shot size, and potential visual-quality issues. The final generation dataset contains 429 samples, consisting of 301 clips from video matting datasets, 43 images from image matting datasets, and 85 Internet videos containing special effects. Although compact, this curated dataset covers challenging transparency phenomena such as sheer fabric, smoke, water, glow, and fine-grained hair boundaries.

Image-to-RGBA Generation Data We first reuse the 429 samples from the text-to-RGBA generation dataset to adapt the pretrained image-to-video model to the RGBA domain, obtaining our base image-to-RGBA model. We then collect an additional set of 196 RGBA effect videos for temporal-pattern adaptation. Each video follows an emergence-disappearance process, in which the effect gradually emerges from a fully transparent state, remains visible in the intermediate frames, and eventually vanishes into transparency. We apply the proposed temporal phase reparameterization to these videos and use them to fine-tune the RGBA first-and-last-frame video model. This compact effect dataset allows the model to learn the target temporal evolution without requiring large-scale middle-frame-conditioned RGBA training data.

4 Experiments

4.1 Experimental Setup

4.1.1 Implementation Details

In Figure 2, the purple dotted line means the specific process for image-to-RGBA model. For text-to-RGBA model, the input is the original video without effect and temporal transformation.

Text-to-RGBA Model Our text-to-RGBA video generator is built upon Wan2.1-T2V-14B [9]. Unless otherwise specified, the model generates 81 frames at a resolution of 480×832 and 16 FPS. We adopt a four-step accelerated sampling schedule without classifier-free guidance. The LoRA rank of both RGB and alpha VAE decoders is 128, while the DoRA rank of the diffusion transformer is 32. The RGBA VAE is trained for 75K iterations with a batch size of 2, and the diffusion transformer is adapted for 1,750 iterations with a batch size of 8. Although the default evaluation setting is 480×832 and 81 frames, the fully convolutional design supports arbitrary aspect ratios, spatial resolutions, and video lengths at 16 FPS.

As shown in Fig. 6, we employ 3D causal convolutional components of Wan-VAE [9] to construct the feature-merging module. The encoded RGB and alpha features are first concatenated and fed into a causal

3D convolution block for t-iteration feature merging, yielding the intermediate representation. Then, we utilize n submodules, which is set as 5, each consisting of a causal residual block and an attention layer to further integrate RGB and alpha features. Finally, a causal 3D convolution predicts the shared RGBA latent Z. We directly predict Z without applying a KL-divergence regularizer, as our objective is to adapt a pretrained latent representation rather than learn a new latent space from scratch.

Image-to-RGBA Model We adopt the Wan2.1-I2V-14B [9] and the RGBA VAE (see Sec. 3.2.1) to construct our image-to-RGBA video generation model, which is first trained for $1,070$ iterations. We utilize the trained RGBA I2V model to initialize the phase-reparameterized model and further fine-tune it on 196 RGBA effect videos for $1,070$ iterations with a batch size of 8. During visibility-aware phase sampling, we set the value of alpha-mass threshold τ_m defined in Eq. (17) to 0.5. During inference, transparent-state guidance is applied to the first 32 of the 40 sampling steps, performing 2–3 gradient iterations per step with a normalized step size $\eta_s = 50$. All image-conditioned videos consist of 81 frames at a resolution of 480×832 and are generated using 40 sampling steps.

Dataset Split Our constructed RGBA dataset consists of $77,237$ and $4,066$ video clips for training and validation of the text-to-RGBA video model, respectively. We further curate 429 samples to train the pretrained image-to-video model, obtaining the base image-to-RGBA video generation model. Then, we additionally collect 196 RGBA effect videos for temporal-pattern adaptation. We respectively generate 60 RGBA videos for the evaluation of text-to-RGBA and image-to-RGBA video generation, respectively. No source asset or temporally overlapping clip is shared between the training and testing sets.

4.1.2 Evaluation Protocol

Text-to-RGBA Video Generation We use VBench [56] to evaluate aesthetic quality, motion smoothness, and temporal flickering. Text alignment and visual naturalness are assessed using GPT-4o [57]. Because these evaluators accept only RGB videos, each RGBA video is composited onto a white background before evaluation. We additionally conduct a perceptive evaluation study for transparency correctness and overall generation quality. Participants rank the outputs of all compared methods, and a lower average rank indicates better performance.

Image-to-RGBA Video Generation We evaluate the image-conditioned video generation models from three complementary perspectives:First, *reference fidelity* measures how well the generated visible frames preserve the input asset. Following DreamBooth [58], we report masked DINO similarity (DINO-I) [59] and masked CLIP image similarity (CLIP-I) [60] between the input asset and generated visible frames. Second, we evaluate general video quality using the same VBench metrics adopted for text-to-RGBA video generation, including aesthetic quality, motion smoothness, and temporal flickering. We compute DINO-I, CLIP-I, and aesthetic quality on frames whose mean alpha over the non-transparent support is greater than 0.5, as appearance-dependent metrics are strongly affected by fully transparent or nearly empty frames; Third, we conduct a perceptive evaluation study to assess temporal-effect correctness and overall video quality.

4.1.3 Compared Methods

Text-to-RGBA Baselines We compare against LayerFlow [30], the combination of LayerDiffuse [5] and AnimateDiff [6], and both the open-source and closed-source versions of TransPixeler [7]. For LayerFlow, we evaluate its foreground-layer generation capability. The closed TransPixeler model is accessed through Adobe Firefly. We additionally compare with a two-stage solution that combines Wan [9] and MatAnyone [61].

Image-to-RGBA Baselines As no end-to-end image-to-RGBA video generation models exist, we construct two-stage baselines by integrating Wan-I2V [9] with MatAnyone [61] for normal RGBA video generation, and Wan2.1-FLF2V [9] with MatAnyone [61] for RGBA effect video generation. Specifically, the input RGBA asset is first composited into an RGB image, animated using Wan-I2V or Wan2.1-FLF2V, and then processed by MatAnyone, with the input alpha channel provided as the initial mask.

4.2 Text-to-RGBA Video Generation

Comparison with Existing Methods Table 1 and Fig. 7 present the quantitative and qualitative comparisons, respectively. LayerDiffuse combined with AnimateDiff exhibits limited motion and weak text alignment; for example, it fails to generate the prompted transparent glass in the third case. This indicates that directly transferring an image-level RGBA representation to a video model is insufficient. LayerFlow better exploits pretrained video priors but produces noticeable object deformation. TransPixeler yields stronger overall results, yet its generated RGB content remains less realistic in challenging fantasy scenes and may contain structural artifacts such as malformed hands. Furthermore, both its open- and closed-source variants produce incorrect glass transparency in Fig. 7, revealing unstable RGB-alpha relationships. In contrast, our method preserves fine structures such as hair, produces realistic fire and smoke, and correctly represents both transparent and opaque glass. These

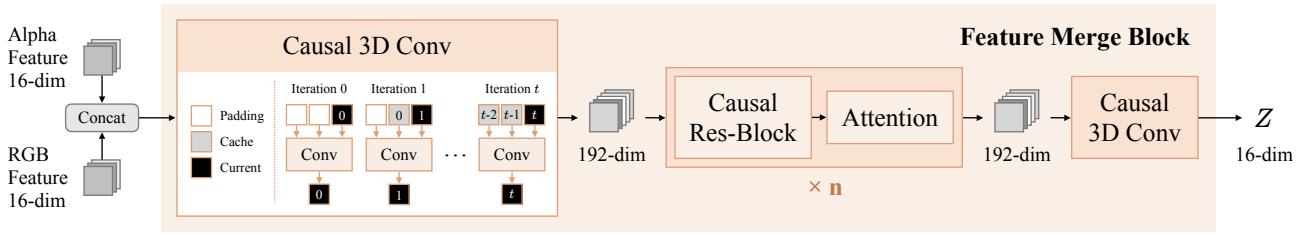


Fig. 6. The detailed architecture for the feature merging block. RGB and alpha features are concatenated and fused by a causal 3D convolution. Then, we use several causal residual blocks and attention layers. Finally, a causal 3D convolution generates the merged latent Z.

visual results are consistent with Table 1: by preserving the pretrained RGB distribution and shifting the alpha distribution, our method achieves the best performance across all reported metrics.

The results of perceptive evaluation study in Table 2 lead to the same conclusion. We conduct online user study to evaluate the transparency correctness and overall quality of generated RGBA videos. We randomly order test cases and show them to 40 participants together with results generated by all other methods. Our method obtains the best average ranks for both transparency correctness and overall quality. Qualitatively, it better preserves fine structures such as hair and produces more natural semi-transparent effects, including smoke, fire, illumination, and glass.

We further compare our method with a two-stage pipeline combining Wan and MatAnyone, which first generates an RGB video with Wan and then extracts the foreground alpha channel using MatAnyone. Because MatAnyone requires a user-provided mask for initialization, the pipeline is not fully automatic. We prompt Wan to generate the subject against a white background to simplify subsequent matting. Nevertheless, as shown in Fig. 8, portions of the white background remain in the resulting RGBA video, leading to inaccurate alpha boundaries and visible artifacts after compositing. More importantly, post-hoc matting assumes a separable foreground and background and therefore cannot reliably recover intrinsically semi-transparent content such as illumination, smoke, and glow. By jointly generating RGB and alpha, our method produces cleaner transparent backgrounds, more accurate fine boundaries, and more faithful semi-transparent effects without requiring an additional mask or matting stage.

Ablation Study of RGBA Representation Learning We evaluate VAE reconstruction quality using PSNR, SSIM [62], and LPIPS [63], latent-distribution separation using RGB-Dist and Alpha-Dist, and text-to-RGBA video generation by disabling various components of RGBA representation learning. Specifically, RGB-Dist and Alpha-Dist measure the Euclidean distances (Dist) [64] between the adapted and pretrained representations for RGB and alpha content, respectively. An effective representation should minimize RGB-Dist while maximizing Alpha-Dist, preserving the pretrained RGB distribution while shifting the alpha distribution. As shown in Table 3, incorporating rendering-based preprocessing and supervision (Rendering) brings significant performance gain in both RGB and alpha reconstruction (see the second row of Table 3), but failing to preserve the pretrained RGB distribution. We further enable transparency-aware bidirectional diffusion (TABD), yielding enhance performance across all evaluation metrics. Our Wan-Alpha incorporates Rendering, TABD, and transparency-guided mean sampler (MS), which further improves text alignment, aesthetics, motion smoothness, and temporal stability without affecting VAE reconstruction quality.

We observe that opaque regions occasionally exhibit holes, as RGB and alpha remain entangled in the shared latent space in the absence of bidirectional diffusion. Our proposed objectives, TABD and MS, separate their roles from the perspective of the frozen DiT, resulting in cleaner foreground alpha maps. Fig. 9 qualitatively demonstrates the effectiveness of TABD and MS. Without both components, the model confuses opaque RGB content with transparency, tending to exhibit holes and incomplete alpha regions within the foreground. Introducing TABD substantially improves the foreground alpha structure by making RGB and alpha representations more distinguishable to the DiT. However, the model may still preserve undesired background content because the shifted transparent latents are more challenging to recover from standard Gaussian noise. The full model addresses this issue using the transparency-guided mean sampler, producing a cleaner transparent background while preserving foreground appearance and fine alpha boundaries.

Ablation Study of Transparency-Guided Sampling Without the transparency-guided mean sampler, the model cannot explicitly control the placement of generated content and often leaves unwanted background artifacts. Mean-shifted noise provides a coarse transparency prior while retaining sufficient flexibility for the model to synthesize fine structures and motion. We use $\mu = 0.05$ by default. A smaller value, such as

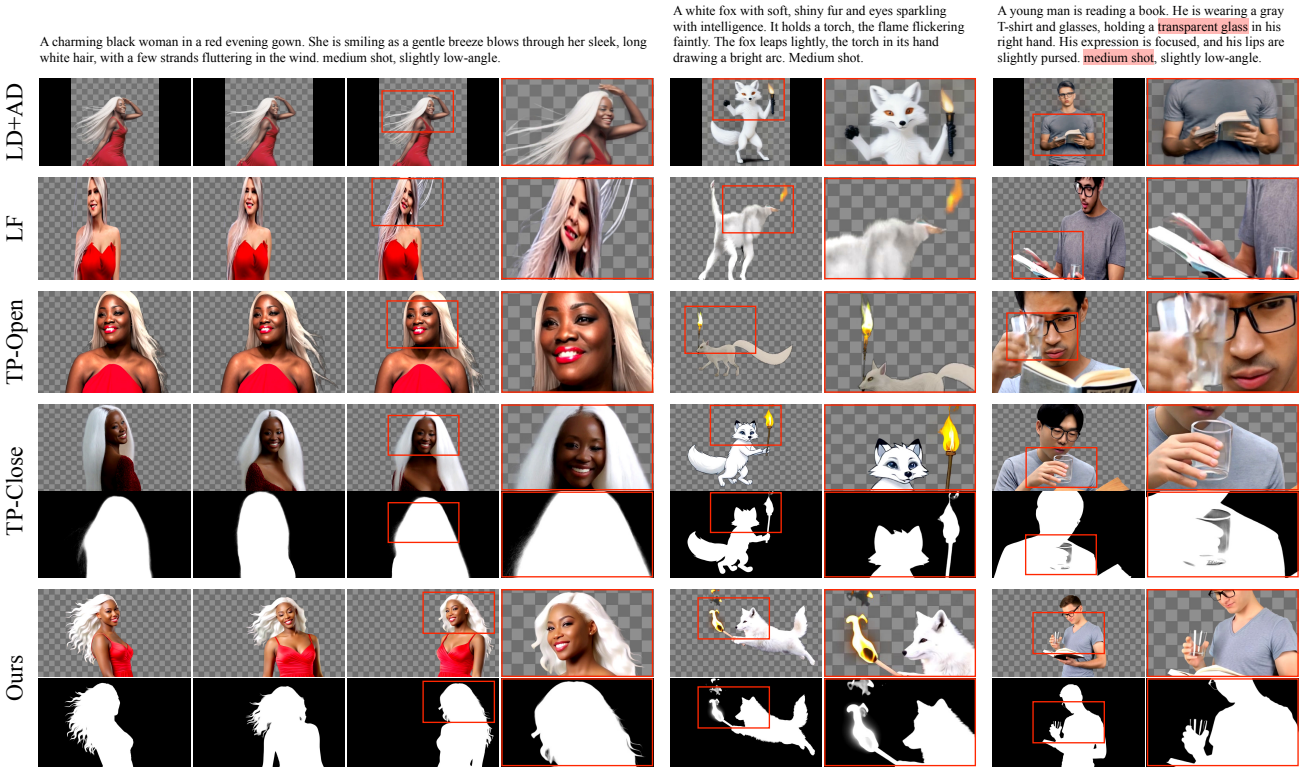


Fig. 7. Qualitative comparison with existing text-to-RGBA video generation methods. Our method produces more realistic RGB content, finer alpha boundaries, and more accurate transparency for challenging objects and semi-transparent effects. Please refer to the supplementary material for video results.

TABLE 1
Quantitative comparison with existing text-to-RGBA video generation methods. Our Wan-Alpha consistently achieves superior video generation performance across all five evaluation metrics.

Method	Text Alignment ↑	Aesthetic Quality ↑	Naturalness↑	Motion Smoothness ↑	Temporal Flickering ↑
LayerFlow (Single) [30]	2.67	0.535	2.35	0.9837	0.9788
LayerDiffuse [5] **+ AnimateDiff** [6]	3.15	0.617	3.03	0.9893	0.9853
TransPixeler (Open) [7]	3.16	0.570	2.97	0.9821	0.9872
TransPixeler (Close) [7]	3.45	0.573	3.07	0.9907	0.9822
Ours	**4.00**	**0.649**	**3.19**	**0.9949**	**0.9941**

TABLE 2
User ranking for text-to-RGBA generation. Lower transparency and overall indicate better performance.

Method	Transparency↓	Overall↓
LayerFlow (Single) [30]	4.29	3.57
LayerDiffuse [5] **+ AnimateDiff** [6]	3.40	4.23
TransPixeler (Open) [7]	2.51	2.71
TransPixeler (Close) [7]	2.57	3.37
Ours	**1.23**	**1.11**

$\mu = 0.03$, provides insufficient transparency control, whereas an excessively large value, such as $\mu = 0.5$, may introduce a slight color bias. Fig. 10 provides a qualitative comparison. With $\mu = 0.03$, the shifted noise provides insufficient transparency guidance and may leave residual background content. The default value $\mu = 0.05$ achieves a favorable balance between transparency control and RGB quality. Increasing the shift to $\mu = 0.5$ excessively perturbs the noise distribution and introduces visible color bias. Moving or rescaling the Gaussian ellipse correspondingly changes the approximate position and size of the generated foreground.

Spatial Transparency Control The Gaussian ellipse mask provides spatial control over the approximate position and extent of generated content. Translating the mask moves the foreground correspondingly, while changing its scale alters the generated object size. Because the mask specifies only a coarse spatial prior, the model retains sufficient flexibility to synthesize fine boundaries, motion, and content-dependent orientation. As visualized in Fig. 11, moving the mask from left to right produces a corresponding displacement of the foreground, demonstrating that the generated alpha support follows the user-specified spatial layout. Enlarging the mask increases the foreground

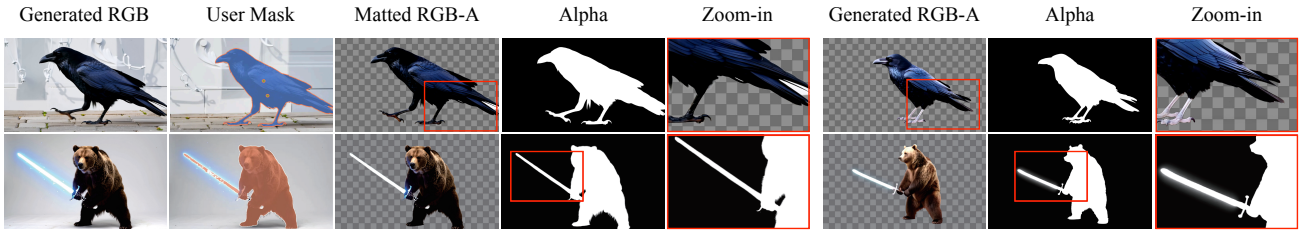


Fig. 8. Qualitative comparison with the two-stage Wan + MatAnyone pipeline. Despite using a white-background prompt and a user-provided initial mask, the matting pipeline leaves residual background content and struggles with semi-transparent effects. Our end-to-end RGBA generation model produces cleaner alpha channels and more accurate transparency.

TABLE 3
Ablation study on RGB-A video reconstruction and generation performance. Our full model achieves the best performance across all metrics, demonstrating the effectiveness of the proposed framework design.

Method			RGB-A VAE						Text-to-Video Generation				
VAE		T2V	PSNR↑		SSIM↑		LPIPS↓		Dist		Text Alignment ↑	Aesthetic Quality ↑	Naturalness↑	Motion Smoothness ↑	Temporal Flickering ↑
Rendering	TABD	MS	RGB	α	RGB	α	RGB	α	RGB↓	α↑					
			40.12	39.98	0.97	0.97	0.043	0.025	0.985	3.56	3.67	0.616	3.17	0.990	0.987
✓			40.88	41.22	0.97	0.98	0.040	0.023	1.015	4.10	3.77	0.619	3.17	0.992	0.984
✓	✓		**41.47**	**42.22**	**0.97**	**0.98**	**0.037**	**0.022**	**0.857**	**4.29**	3.86	0.621	3.19	0.992	0.990
✓	✓	✓	**41.47**	**42.22**	**0.97**	**0.98**	**0.037**	**0.022**	**0.857**	**4.29**	**4.00**	**0.649**	**3.19**	**0.995**	**0.994**

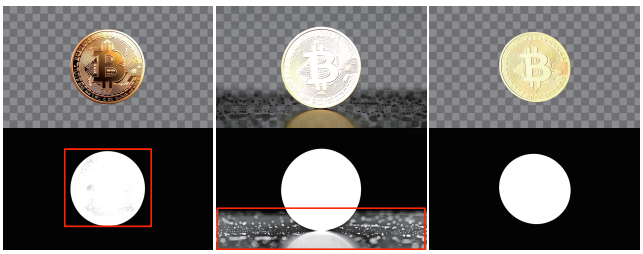


Fig. 9. Qualitative ablation of the text-to-RGBA generation framework. TABD improves RGB–alpha separation, while MS facilitates clean transparent backgrounds.



Fig. 10. Effect of the mean-shift strength μ in the transparency-guided mean sampler.

extent rather than merely stretching its appearance. Meanwhile, the model automatically adjusts object pose and orientation to maintain a natural composition under different mask configurations. These results show that the Gaussian mask controls coarse transparency geometry without over-constraining fine structures or temporal dynamics. Additional qualitative examples confirm this behavior across human subjects, animals, smoke, fire, and illumination effects.

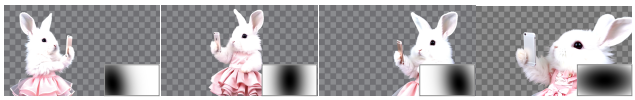


Fig. 11. Results under different Gaussian ellipse masks G (shown at the bottom right). Users can use G to control the shape and position of transparent regions.

4.3 Image-to-RGBA Video Generation

Image-to-RGBA video generation remains largely unexplored. To the best of our knowledge, no existing method directly animates a user-provided RGBA asset while jointly generating its RGB appearance and temporally coherent alpha channel. Therefore, we construct two-stage baselines by integrating Wan2.1-I2V with MatAnyone for normal first-frame-conditioned RGBA video generation, and Wan2.1-FLF2V with MatAnyone for emergence–disappearance RGBA effect video generation, where arbitrary-frame conditioning enables alpha propagation from designated visible reference frames. The latter is more challenging because the foreground becomes fully transparent during the sequence, requiring the model to synthesize a complete transition between visible and transparent states while preserving the reference appearance. For both normal and RGBA effect video generation, we compare our direct RGBA video generation models against the corresponding two-stage generation baselines in terms of *reference fidelity* and overall video generation quality.

Comparison with Competing Normal RGBA Video Generation Baselines As reported in Table 4, our method consistently outperforms the two-stage generation-and-matting pipelines across all metrics. For normal RGBA videos, our method improves DINO-I from 0.9033 to 0.9066 and CLIP-I from 0.9089 to 0.9445, demonstrating better preservation of the structure, semantics, and appearance of the input asset. It also achieves slightly higher aesthetic quality, motion smoothness, and temporal flickering scores. Since both methods are built upon Wan2.1-I2V and thus share similar RGB generation and motion priors, these consistent improvements indicate that directly modeling RGB and alpha jointly can mitigate the information loss and propagation errors introduced by post-hoc video matting.

We present qualitative comparisons for normal RGBA video generation in Fig. 12. Two-stage baseline, Wan+MatAnyone, first uses Wan2.1-I2V to render the asset into an RGB video, where transparency information is irreversibly mixed with the composited background. MatAnyone then relies on an interactively specified segmentation mask to initialize the matting process. This two-stage pipeline inevitably introduces and accumulates errors, particularly around thin structures and low-opacity boundaries. These errors are subsequently propagated to the estimated alpha channel, resulting in overly opaque regions or the loss of soft transparency gradients. In contrast, our proposed model achieves superior *reference fidelity* by jointly modeling RGB appearance and alpha boundaries throughout the video generation process. More fundamentally, post-hoc matting cannot reliably recover intrinsic transparency once semi-transparent foreground colors have been blended with the background in the generated RGB frames. It therefore tends to produce nearly opaque alpha values or lose soft transparency gradients. Beyond these quality limitations, the two-stage pipeline requires separate RGB generation, interactive mask specification, and video matting, making the overall process cumbersome and labor-intensive for practical applications. In contrast, our method directly generates RGB and alpha channels from the input RGBA asset, eliminating the need for additional user interaction.

Comparison with RGBA Effect Video Generation Baselines We present quantitative comparisons with the competing baseline in Table 4 and qualitative comparisons in Fig. 13. Our proposed framework achieves significant improvements in both DINO-I and CLIP-I over the two-stage Wan2.1-FLF2V+MatAnyone baseline. These performance gains demonstrate that temporal phase reparameterization effectively utilizes the visible conditioning frame to anchor both the structural identity and semantic appearance of the generated effect. Furthermore, our framework consistently achieves superior overall video quality in terms of aesthetic quality, motion smoothness, and temporal flickering. This demonstrates that explicitly learning the transition between visible and transparent states produces more natural content and more temporally coherent alpha evolution than propagating an interactively specified mask after RGB video generation.

Fig. 13 further reveals the limitations of applying post-hoc video matting to emergence–disappearance effects. This limitation becomes more pronounced for emergence–disappearance RGBA effect video generation. Although MatAnyone supports arbitrary-frame mask conditioning, it primarily propagates foreground masks and does not explicitly model the transition between positive- and zero-alpha states. Therefore, it tends to leave residual alpha during disappearance or produce temporally unstable boundaries during reappearance.

We investigate two possible initialization positions for the two-stage Wan2.1-FLF2V+MatAnyone pipeline. When MatAnyone is initialized from the first frame (*First*), its interactive mask can provide a complete foreground at the initial frame. However, requiring the first frame to remain visible is inherently conflicted with the desired emergence-disappearance RGBA effect, in which the content is expected to emerge from a transparent state. Therefore, this setting cannot produce the complete transparent-visible-transparent results. Moreover, errors introduced during RGB video generation (see distorted text shown in Fig. 13), are propagated to the subsequent matting stage and cannot be effectively corrected afterward.

An alternative is to initialize MatAnyone from a visible middle frame (*Middle*). Although the interactive mask successfully extracts the object at the condition frame, MatAnyone performs forward propagation from this initialization frame. Therefore, the frames preceding the middle condition are discarded, yielding only an incomplete second half of the RGBA video. Moreover, when the object subsequently disappears from the RGB frames, the propagated matting memory continues to predict a foreground region, resulting in a persistent object-shaped artifact in the alpha channel. This limitation highlights that tracking-oriented video matting does not explicitly model transitions to a fully transparent state.

In contrast, our method treats the visible condition frame as an appearance anchor while explicitly controlling the transparent intervals through temporal phase reparameterization and transparent-state guidance. This design enables our proposed framework to generate the complete emergence–disappearance process, preserve the conditioned appearance, and produce clean zero-alpha frames after the effect vanishes.

Perceptive Evaluation Study We conduct a per-

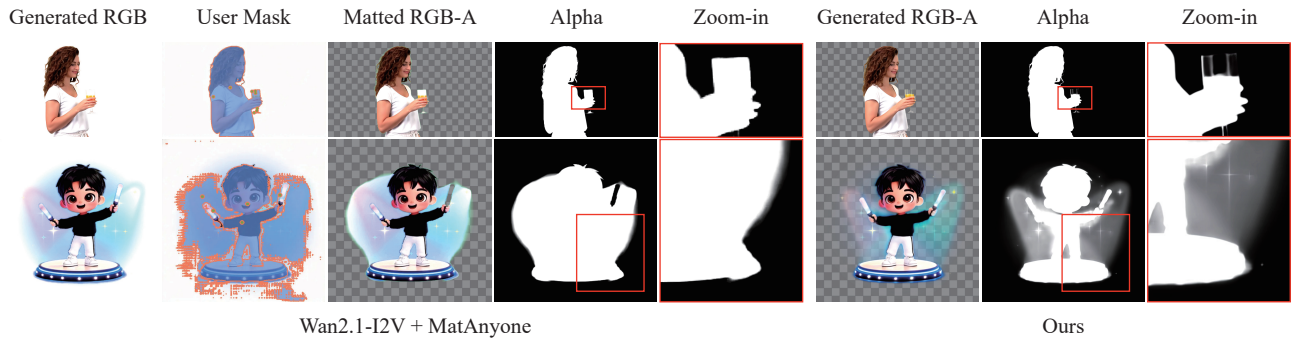


Fig. 12. Qualitative comparison with the two-stage Wan2.1-I2V + MatAnyone pipeline. Despite using a white-background prompt and a user-provided initial mask, the matting pipeline leaves residual background content and struggles with semi-transparent effects. Our end-to-end RGBA generation model produces cleaner alpha channels and more accurate transparency.

TABLE 4
Quantitative comparison on normal and effect image-to-RGBA video generation. DINO-I and CLIP-I measure reference fidelity, while the remaining metrics evaluate general video quality.

Method		DINO-I↑	CLIP-I↑	Aesthetic Quality↑	Motion Smoothness↑	Temporal Flickering↑
Normal	**Wan2.1-I2V + MatAnyone** [61]	0.9033	0.9089	0.6296	0.9912	0.9858
	Ours	**0.9066**	**0.9445**	**0.6305**	**0.9916**	**0.9864**
Effect	**Wan2.1-FLF2V + MatAnyone** [61]	0.6679	0.7969	0.5749	0.9917	0.9926
	Ours	**0.9227**	**0.9368**	**0.5968**	**0.9941**	**0.9930**

TABLE 5
User ranking for image-to-RGBA generation.

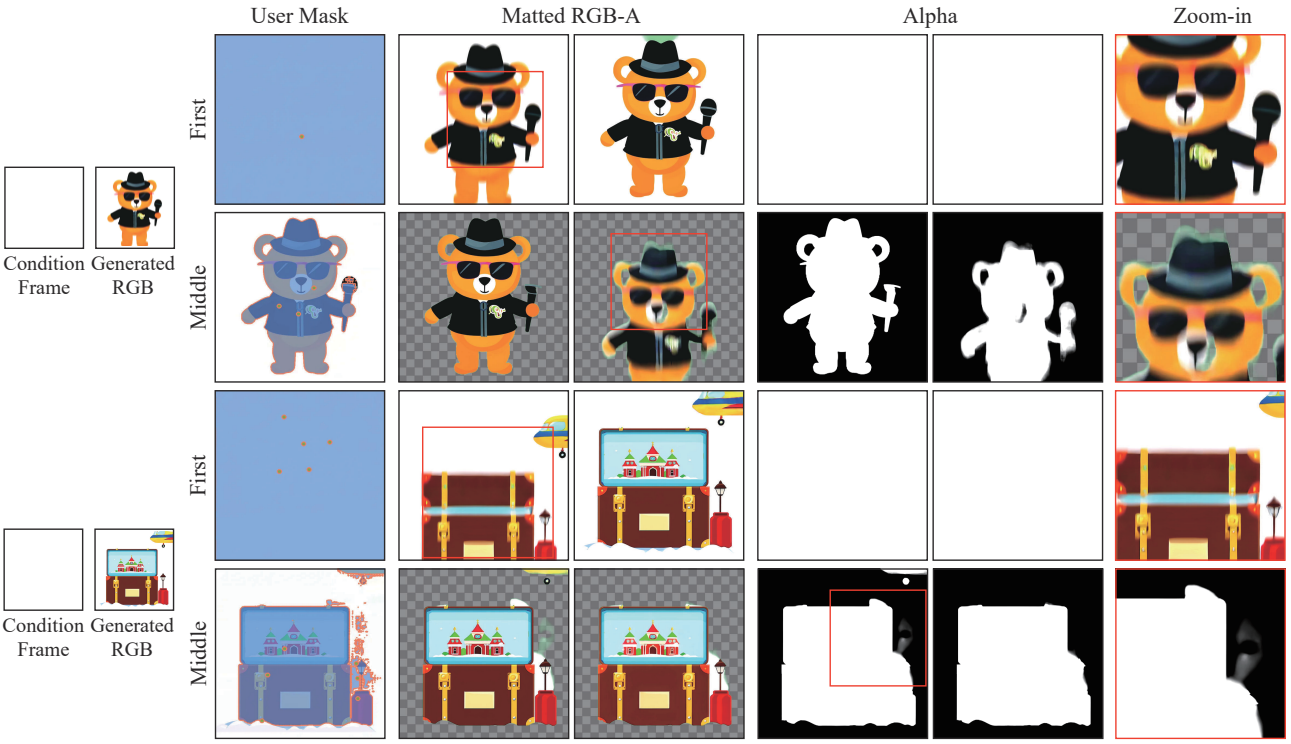
Method		Temporal Effect↓	Overall↓
Normal	**Wan2.1-I2V + MatAnyone**	1.73	1.62
	Ours	**1.27**	**1.38**
Effect	**Wan2.1-FLF2V + MatAnyone**	1.88	1.84
	Ours	**1.12**	**1.16**

ceptive evaluation study using randomly ordered test cases. 40 participants are shown the input RGBA asset together with videos generated by all methods and are asked to rank them according to temporal-effect correctness and overall video quality. Our method achieves better results across all metrics.

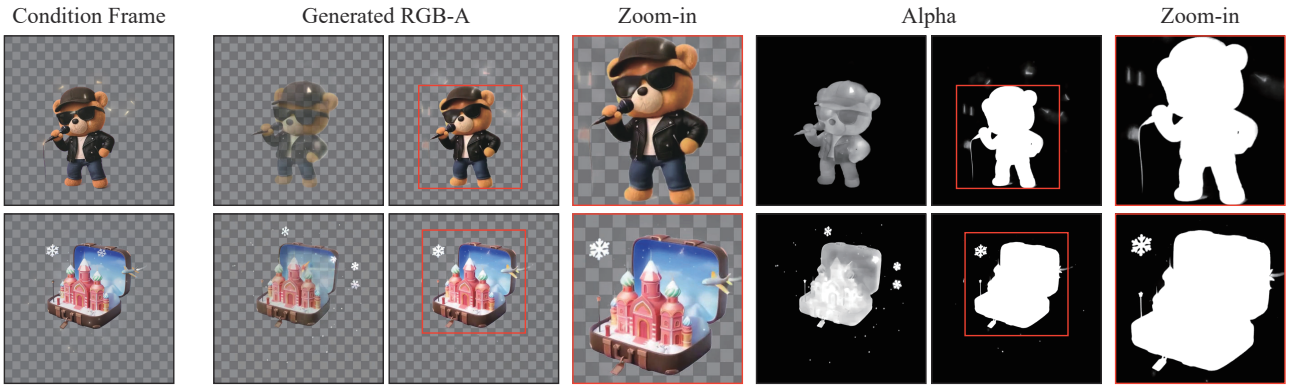
Ablation Study Our image-to-RGBA effect generation framework contains two principal designs: temporal phase reparameterization and transparent-state guidance. We examine the effect of removing each component, with the results reported in Table 6. We first remove both components, degrading the model into the Wan2.1-FLF2V-based RGBA video generation model that uses effect videos in their original temporal order. Since the first and last frames of these videos are fully transparent, the boundary conditions contain neither observable RGB appearance nor spatial alpha structure. Consequently, the image-conditioning mechanism becomes ineffective and degrades into text-to-RGBA video generation, resulting in low similarity between the input asset and generated visible video frames (see the first row of Table 6).

We then incorporate temporal phase reparameterization, which transforms the original zero-positive-zero sequence into a positive-zero-positive sequence. The resulting visible boundary frames provide effective appearance conditions, substantially improving *reference fidelity* (see the second row of Table 6). However, temporal phase reparameterization alone only enables the model to learn the distribution of disappearance-emergence RGBA video effects. With limited training data, this learned distribution is insufficient to reliably determine the location and duration of the transparent interval during inference. Finally, we apply transparent-state guidance during inference to explicitly constrain the user-specified interval toward transparency. Combined with temporal phase reparameterization, the full Wan-Alpha++ preserves the conditioned appearance while generating a stable and controllable emergence-disappearance process, achieving the best overall performance.

Sensitivity to Guidance Hyperparameters We change the guidance step size and the number of guided sampling steps to examine their effects on video generation performance, with the results reported in Table 7. A larger step size (e.g., $\eta_s = 50$) provides stronger control over the transparent state, leading to consistent improvements across all metrics. However, an excessively large step size (e.g., $\eta_s = 75$) overly perturbs the generated appearance, resulting in degraded performance. For the guidance duration, applying the guidance step for 16 or 24 slightly improves *reference fidelity* in terms of DINO-I and CLIP-I, but results in less stable temporal RGBA video effects. Extending guidance to the first 32 of 40 sam-



(a) Wan2.1-FLF2V + MatAnyone



(b) Ours

Fig. 13. Qualitative comparison with the two-stage Wan2.1-FLF2V + MatAnyone pipeline under different matting initialization positions. With *First* initialization, MatAnyone extracts the visible first frame, but fails to produce the emergence-disappearance RGBA effect. The RGB generator also fails to preserve fine appearance details such as text. With *Middle* initialization, MatAnyone only processes the frames following the condition frame and discards the first half of the video. Moreover, it continues to propagate an object-shaped alpha artifact even after the RGB content has disappeared. In contrast, our method conditions on a user-specified visible frame and generates a complete RGBA effect that gradually emerges and then disappears with clean transparent states.

pling steps achieves the best overall video generation performance across aesthetic quality, motion smoothness, and temporal consistency. However, applying guidance throughout all 40 steps introduces excessive cumulative latent updates, degrading details in visible frames. Therefore, we adopt $\eta_s = 50$ and 32 guided sampling steps as the default configuration.

5 Conclusion

In this work, we present Wan-Alpha++, a unified framework for high-quality and controllable RGBA video generation. We first address the entanglement between RGB appearance and alpha transparency by learning shiftable RGB–alpha distributions in both latent and noise spaces. The proposed transparency-aware bidirectional diffusion objective preserves the pretrained RGB distribution while making alpha representations more distinguish-

TABLE 6
Component ablation of temporal phase reparameterization (**Reparameterization**) and transparent-state guidance (**Guidance**). We report reference fidelity, general video quality, and the user ranking for temporal-effect correctness. Lower is better for Temporal Effect, while higher is better for all other metrics.

Reparameterization	Guidance	DINO-I ↑	CLIP-I ↑	Aesthetic Quality ↑	Motion Smoothness ↑	Temporal Flickering ↑	Temporal Effect ↓
		0.6315	0.7542	0.5793	0.9919	0.9918	2.76
✓		0.9136	0.9275	0.5924	0.9936	0.9927	2.09
✓	✓	**0.9227**	**0.9368**	**0.5968**	**0.9941**	**0.9930**	**1.12**

TABLE 7
Sensitivity to transparent-state guidance hyperparameters.

Parameter	Value	DINO-I↑	CLIP-I↑	Aesthetic Quality↑	Motion Smoothness↑	Temporal Flickering↑
Step size η_s	10	0.9208	0.9335	0.5908	0.9935	0.9925
	25	0.9220	0.9352	0.5947	0.9939	0.9928
	50	**0.9227**	**0.9368**	**0.5968**	**0.9941**	**0.9930**
	75	0.9188	0.9310	0.5921	0.9934	0.9924
Guided sampling steps	16	**0.9235**	**0.9372**	0.5905	0.9934	0.9924
	24	0.9230	0.9369	0.5943	0.9938	0.9927
	32	0.9227	0.9368	**0.5968**	**0.9941**	**0.9930**
	40	0.9175	0.9295	0.5897	0.9931	0.9920

able, and the transparency-guided mean sampler improves transparency synthesis while providing coarse spatial control over the generated content. We further extend RGBA video generation from text conditioning to practical image conditioning. For emergence-disappearance effects, fully transparent boundary frames invalidate conventional image-to-video conditioning. Temporal phase reparameterization addresses this problem by moving visible interior content to the temporal boundaries, allowing the appearance-conditioning prior of a pretrained FLF2V model to be reused with limited RGBA data. Transparent-state guidance subsequently controls the location and duration of the invisible interval, and inverse phase transformation restores the desired emergence-disappearance process. Together, these designs provide complementary spatial and temporal control over RGBA generation. Experiments demonstrate that Wan-Alpha++ improves RGB quality, reference fidelity, transparency rendering, motion smoothness, and temporal consistency over existing RGBA generation and two-stage video-matting pipelines. The generated assets can be directly composited onto arbitrary backgrounds, supporting applications in visual effects, filmmaking, game development, and digital content creation. Future work may extend the proposed temporal reparameterization to more complex multi-stage transparency events and reduce the additional inference cost introduced by gradient guidance.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No.62476192). It was also funded by Laboratory for Artificial Intelligence in Design (Project 3.1), the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

REFERENCES

[1] X. Chen, Z. Chen, and Y. Song, "Transanimate: Taming layer diffusion to generate rgba video," *ArXiv preprint*, 2025.
[2] M. Li, Z. Zhang, J. Liao, L. Qin, and W. Wang, "Transvdm: Motion-constrained video diffusion model for transparent video synthesis," in *ICASSP*, 2025.
[3] J. Bai, J. Zhou, B. Wang, W. Chen, Y. Yang, Z. Lei, and F. Wang, "Layer-animate for transparent video generation," in *ICASSP*, 2025.
[4] T. Zhang, Z. Yuan, Y. Zhu, J. Zhou, and J. Zhang, "Ildiff: Generate transparent animated stickers by implicit layout distillation," in *ICASSP*, 2025.
[5] L. Zhang and M. Agrawala, "Transparent image layer diffusion using latent transparency," *ACM TOG*, 2024.
[6] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai, "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning," in *ICLR*, 2024.
[7] L. Wang, Y. Li, Z. Chen, J.-H. Wang, Z. Zhang, H. Zhang, Z. Lin, and Y.-C. Chen, "Transpixeler: Advancing text-to-video generation with transparency," in *CVPR*, 2025.
[8] H. Dong, W. Wang, C. Li, J. Lyu, and D. Lin, "Video generation with stable transparency via shiftable rgb-a distribution learner," in *CVPR*, 2026.
[9] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, *et al.*, "Wan: Open and advanced large-scale video generative models," *ArXiv preprint*, 2025.
[10] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, *et al.*, "Hunyuanvideo: A systematic framework for large video generative models," *ArXiv preprint*, 2024.
[11] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, D. Yin, Y. Zhang, W. Wang, Y. Cheng, B. Xu, X. Gu, Y. Dong, and J. Tang, "Cogvideox: Text-to-video diffusion models with an expert transformer," in *ICLR*, 2025.
[12] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, G. Liu, A. Raj, *et al.*, "Lumiere: A space-time diffusion model for video generation," in *SIGGRAPH Asia*, 2024.
[13] X. Shi, Z. Huang, F.-Y. Wang, W. Bian, D. Li, Y. Zhang, M. Zhang, K. C. Cheung, S. See, H. Qin, *et al.*, "Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling," in *SIGGRAPH*, 2024.

[14] X. Li, D. Lin, S. Gao, J. Li, R. Liu, and Q. Guo, "Open-vocabulary part segmentation via progressive and boundary-aware strategy," in *NeurIPS*, 2025.

[15] D. Lin, W. Xin, J. Shen, R. Zhang, R. Liu, M. Wang, W. Xie, Q. Guo, and P. Li, "Generative status estimation and information decoupling for image rain removal," *NeurIPS*, 2022.

[16] X. Wang, D. Lin, W. Su, J. Du, R. Zhang, J. Zhang, H. Dong, K. Xu, Q. Guo, and P. Li, "Hrc-net: Learning visual hypothesis, representative, and collaboration for multi-domain image inpainting," *ACM TOG*, 2025.

[17] H. Dong, E. Ma, L. Wang, M. Wang, W. Xie, Q. Guo, P. Li, L. Liang, K. Yang, and D. Lin, "Cvsformer: Cross-view synthesis transformer for semantic scene completion," in *ICCV*, 2023.

[18] X. Duan, Y. Cao, L. Zhu, G. Fu, X. Wang, R. Zhang, and P. Li, "Two-stage video shadow detection via temporal-spatial adaption," in *ECCV*, 2024.

[19] K. Xu, X. Tian, X. Yang, B. Yin, and R. W. H. Lau, "Intensity-aware single-image deraining with semantic and color regularization," *IEEE TIP*, 2021.

[20] Y. Li, K. Xu, G. P. Hancke, and R. W. Lau, "Color shift estimation-and-correction for image enhancement," in *CVPR*, 2024.

[21] K. Xu, G. P. Hancke, and R. W. H. Lau, "Learning image harmonization in the linear color space," in *ICCV*, 2023.

[22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[23] Z. Wang, H. Yu, J. Zhan, and C. Yuan, "Alphavae: Unified end-to-end RGBA image reconstruction and generation with alpha-aware representation learning," *ArXiv preprint*, 2025.

[24] F. Quattrini, V. Pippi, S. Cascianelli, and R. Cucchiara, "Alfie: Democratising rgba image generation with no," in *ECCV*, 2024.

[25] J. Huang, P. Yan, J. Cai, J. Liu, Z. Wang, Y. Wang, X. Wu, and G. Li, "Dreamlayer: Simultaneous multi-layer generation via diffusion mode," *ArXiv preprint*, 2025.

[26] J. Yang, Q. Liu, Y. Li, S. Y. Kim, D. Pakhomov, M. Ren, J. Zhang, Z. Lin, C. Xie, and Y. Zhou, "Generative image layer decomposition with visual effects," in *CVPR*, 2025.

[27] D. Huang, W. Li, Y. Zhao, X. Pan, Y. Zeng, and B. Dai, "Psdiffusion: Harmonized multi-layer image generation via layout and appearance alignment," *ArXiv preprint*, 2025.

[28] K. Kang, G. Sim, G. Kim, D. Kim, S. Nam, and S. Cho, "Layeringdiff: Layered image synthesis via generation, then disassembly with generative knowledge," *ArXiv preprint*, 2025.

[29] Y. Dai, H. Li, S. Zhou, and C. C. Loy, "Trans-adapter: A plug-and-play framework for transparent image inpainting," *ArXiv preprint*, 2025.

[30] S. Ji, H. Luo, X. Chen, Y. Tu, Y. Wang, and H. Zhao, "Layerflow: A unified model for layer-aware video generation," in *ACM SIGGRAPH*, 2025.

[31] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *ArXiv preprint*, 2023.

[32] J. Xing, M. Xia, Y. Zhang, H. Chen, W. Yu, H. Liu, G. Liu, X. Wang, Y. Shan, and T.-T. Wong, "Dynamicrafter: Animating open-domain images with video diffusion priors," in *ECCV*, 2024.

[33] Y. Wang, Z. Chen, X. Chen, Y. Wei, J. Zhu, and J. Chen, "Framebridge: Improving image-to-video generation with bridge models," in *ICML*, 2026.

[34] Y. Zeng, G. Wei, J. Zheng, J. Zou, Y. Wei, Y. Zhang, and H. Li, "Make pixels dance: High-dynamic video generation," in *CVPR*, 2024.

[35] X. Chen, Y. Wang, L. Zhang, S. Zhuang, X. Ma, J. Yu, Y. Wang, D. Lin, Y. Qiao, and Z. Liu, "Seine: Short-to-long video diffusion model for generative transition and prediction," in *ICLR*, 2023.

[36] J. Xing, H. Liu, M. Xia, Y. Zhang, X. Wang, Y. Shan, and T.-T. Wong, "Tooncrafter: Generative cartoon interpolation," *ACM TOG*, 2024.

[37] S. Yang, T. Kwon, and J. C. Ye, "Vibidsampler: Enhancing video interpolation using bidirectional diffusion sampler," in *ICLR*, 2025.

[38] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, "Freedom: Training-free energy-guided conditional diffusion model," in *ICCV*, 2023.

[39] Y. He, N. Murata, C.-H. Lai, Y. Takida, T. Uesaka, D. Kim, W. Liao, Y. Mitsufuji, Z. Kolter, R. Salakhutdinov, *et al.*, "Manifold preserving guided diffusion," in *ICLR*, 2024.

[40] Z. Xiao, Y. Zhou, S. Yang, and X. Pan, "Video diffusion models are training-free motion interpreter and controller," *NeurIPS*, 2024.

[41] C. Simon, M. Ishii, A. Hayakawa, Z. Zhong, S. Takahashi, T. Shibuya, and Y. Mitsufuji, "Titan-guide: Taming inference-time alignment for guided text-to-video diffusion models," in *ICCV*, 2025.

[42] B. Lai, S. Lee, X. Cao, X. Li, and J. M. Rehg, "Unified text-image-to-video generation: A training-free approach to flexible visual conditioning," *ArXiv preprint*, 2025.

[43] J. Li, J. Zhang, and D. Tao, "Deep automatic natural image matting," in *IJCAI*, 2021.

[44] J. Li, J. Zhang, S. J. Maybank, and D. Tao, "Bridging composite and real: towards end-to-end deep image matting," *IJCV*, 2022.

[45] Y. Qiao, Y. Liu, X. Yang, D. Zhou, M. Xu, Q. Zhang, and X. Wei, "Attention-guided hierarchical structure aggregation for image matting," in *CVPR*, 2020.

[46] Y. Sun, C.-K. Tang, and Y.-W. Tai, "Ultrahigh resolution image/video matting with spatio-temporal sparsity," in *CVPR*, 2023.

[47] Y. Liu, J. Xie, X. Shi, Y. Qiao, Y. Huang, Y. Tang, and X. Yang, "Tripartite information mining and integration for image matting," in *ICCV*, 2021.

[48] S. Ma, J. Li, J. Zhang, H. Zhang, and D. Tao, "Rethinking portrait matting with pirvacy preserving," *IJCV*, 2023.

[49] Q. Yu, J. Zhang, H. Zhang, Y. Wang, Z. Lin, N. Xu, Y. Bai, and A. Yuille, "Mask guided matting via progressive refinement network," in *CVPR*, 2021.

[50] Y. Sun, C.-K. Tang, and Y.-W. Tai, "Semantic image matting," in *CVPR*, 2021.

[51] H. Cai, F. Xue, L. Xu, and L. Guo, "Transmatting: Enhancing transparent objects matting with transformers," in *ECCV*, 2022.

[52] Y. Sun, G. Wang, Q. Gu, C.-K. Tang, and Y.-W. Tai, "Deep video matting via spatio-temporal alignment and aggregation," in *CVPR*, 2021.

[53] Y. Zhang, C. Wang, M. Cui, P. Ren, X. Xie, X.-S. Hua, H. Bao, Q. Huang, and W. Xu, "Attention-guided temporally coherent video object matting," in *ACM MM*, 2021.

[54] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Real-time high-resolution background matting," in *CVPR*, 2021.

[55] Q. Team, "Qwen2.5-vl." https://qwenlm.github.io/blog/qwen2.5-vl/, 2025.

[56] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu, "VBench: Comprehensive benchmark suite for video generative models," in *CVPR*, 2024.

[57] OpenAI, "GPT-4 technical report," *ArXiv preprint*, 2023.

[58] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *CVPR*, 2023.

[59] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[60] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[61] P. Yang, S. Zhou, J. Zhao, Q. Tao, and C. C. Loy, "Matanyone: Stable video matting with consistent memory propagation," in *CVPR*, 2025.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, 2004.

[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[64] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, 2015.