ALPHA AS AN EFFICIENCY SIGNAL: VISIBILITY-ROUTED RGBA IMAGE-TO-VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

RGBA videos combine RGB appearance with an alpha channel, enabling animated assets to be applied across arbitrary backgrounds, which are heavily used in gaming industry. However, generating high-quality RGBA animations for games remains challenging for two reasons. First, most existing RGBA video datasets are dominated by photorealistic content, with limited coverage of game assets. Second, the traditional generate-then-matte pipelines estimate alpha only after RGB synthesis, so semi-transparent regions are often blurred by background, resulting in unstable matting outputs. More recently, many methods have begun to model RGB and alpha jointly, but existing approaches are mostly text-conditioned, and still have unresolved issues in efficiency and quality. To address these challenges, we introduce **GameAlpha-2.4K**, a game-style RGBA video dataset built with matte-friendly synthesis, multi-hypothesis alpha recovery, and compositing-based quality gates. Using this dataset, we train a reference-conditioned RGBA video generator that jointly produces RGB frames and alpha mattes in a single pass. To improve efficiency, we propose a **visibility router** that identifies transparent tokens in an early stage and bypasses their later DiT updates, while x_0-lock guides them along the original flow-matching schedule toward self-predicted endpoints. Our model obtains lower FVD than fine-tuned two-stage pipelines. Relative to our dense generator, visibility routing reduces mean end-to-end runtime from 125.9 to 104.8 seconds per video ($1.20\times$), with near-dense quality.

1 INTRODUCTION

We study reference-conditioned RGBA image-to-video generation. Given a single RGBA reference image and a text prompt, the model jointly generates foreground color and a temporally coherent alpha matte for subsequent compositing onto different backgrounds.

Existing solutions fall into two families, each leaving part of the problem unresolved. The first generates an RGB video and then estimates alpha with segmentation or video matting (Ravi et al., 2024; Yang et al., 2025; Lim et al., 2026). This decomposition is often limited for semi-transparent content: once background color has mixed into a pixel, estimating alpha alone is usually insufficient to recover a clean foreground for re-compositing; hard segmentation discards partial opacity outright, and matting tends to flicker around hair, glow, and motion blur—failures that concentrate exactly in the soft regions on which compositable assets are judged. The second generates RGBA directly and avoids the decomposition, but open reference-conditioned models remain scarce: image-level methods carry no motion, open video models are mostly text-to-video, and image-to-video attempts are domain-specific or not fully released (Wang et al., 2025; Dong et al., 2025; Zhang et al., 2026; Zhang & Agrawala, 2024; Chen et al., 2026b; Li et al., 2025). How alpha is represented further sets the cost of direct generation: TransPixeler appends alpha tokens and TransText tiles alpha beside RGB, both inflating the DiT sequence, whereas Wan-Alpha folds RGB and alpha into one merged latent at no token overhead—the representation we adopt. Yet all existing direct generators, merged-latent designs included, spend DiT computation uniformly over spatiotemporal tokens, leaving the visibility information carried by alpha unused.

RGBA compositing provides an analytic visibility signal. Given $C = \alpha R + (1-\alpha)B$, $\frac{\partial C}{\partial R} = \alpha$; conditioned on a region's final alpha being zero, further refinement of its RGB layer cannot affect any composite. Final transparency is unknown during denoising, and alpha errors remain consequential. The challenge is to predict final-transparent tokens early enough to reduce computation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We use this observation as an inference-time routing principle while designing the model to generate RGB and alpha jointly. To address the lack of reference-conditioned RGBA data for game-asset animation, we construct GameAlpha-2.4K, whose alpha mattes are recovered from videos generated under controlled, matte-friendly conditions and retained through staged quality checks. For generation, we adapt a merged RGB-A representation to the pretrained model's native image-to-video conditioning. The reference RGB and alpha are fused into a first-frame latent anchor, while its white-background composite and text enter the model's visual and textual conditioning pathways. A single DoRA-adapted DiT then denoises the joint RGB-A latent without introducing additional DiT tokens or an alpha-specific DiT branch.

A visibility router reads the model's step-2 clean estimate and predicts which tokens will remain fully transparent. After retaining a conservative spatiotemporal margin around the predicted foreground, the router advances the routed tokens along an x_0-locked flow path toward their step-2 predicted endpoints. Subsequent sparse steps reactivate tokens near the evolving foreground boundary, while low-support or high-uncertainty samples remain on the dense path. The router preserves the denoising schedule and backbone parameters; we evaluate it with four-step distillation.

Our contributions are:

- **Task and dataset.** We study reference conditioned RGBA image-to-video generation for compositable game-asset animation and build GameAlpha-2.4K: 2,395 clips with alpha mattes are recovered under controlled, matte-friendly conditions and verified by staged quality gates.
- **A visibility-aware framework.** We introduce a visibility router with x_0-lock that predicts effectively transparent tokens from an early clean estimate and safely reduces their late-step computation using a spatiotemporal margin and a pre-routing dense fallback.
- **Empirical validation.** On GameAlpha-2.4K, our single-stage model lowers FVD versus fine-tuned same-backbone generate-then-matte baselines; with an average initial routed fraction of 35% after evaluation 2, reactivation-aware routing achieves a $1.20\times$ end-to-end speedup over our dense model with near-dense metrics and the lowest FVD among compute-matched token-selection baselines.

2 Related Work

2.1 RGBA and Layered Generation

RGBA and layered generation model appearance together with explicit opacity, producing foregrounds or layers that can be composited independently. At the image level, LayerDiffuse (Zhang & Agrawala, 2024) encodes transparency in a frozen diffusion latent and ART (Pu et al., 2025) extends it to multiple layers, but neither models motion. Video methods include text-conditioned RGBA or layered generation with TransPixeler (Wang et al., 2025), Wan-Alpha (Dong et al., 2025), and LayerFlow (Ji et al., 2025). Reference-conditioned or specialized I2V methods include TransText (Zhang et al., 2026) for glyph animation, TransVDM (Li et al., 2025) for image-and-text-conditioned synthesis, and motion-guided TransAnimate (Chen et al., 2026b). TransPixeler also demonstrates CogVideoX-based I2V, but no corresponding checkpoint or I2V code is public. Cost separates these representations: TransPixeler and TransText enlarge the token sequence or spatial canvas, increasing attention cost, whereas Wan-Alpha folds RGB and alpha into a merged latent without token overhead. We combine Wan-Alpha's merged latent with native I2V conditioning and use the generated alpha to route inference computation.

2.2 Video Matting and Segmentation

Two-stage pipelines generate RGB before estimating a segmentation mask or alpha matte. SAM2/3 produce hard masks without partial opacity (Ravi et al., 2024; Carion et al., 2025), while Matting-Anything (Li et al., 2023) estimates soft alpha at the image level. Video matting models use temporal recurrence, memory, or global aggregation (Lin et al., 2021b; Li et al., 2022; Yang et al., 2025; 2026), with VideoMaMa (Lim et al., 2026) further introducing a generative video prior. Despite these advances, matting remains post-hoc: a composite underdetermines foreground and alpha, so estimating alpha alone after RGB synthesis cannot guarantee a background-independent foreground layer. We instead generate paired foreground color and alpha in one pass.

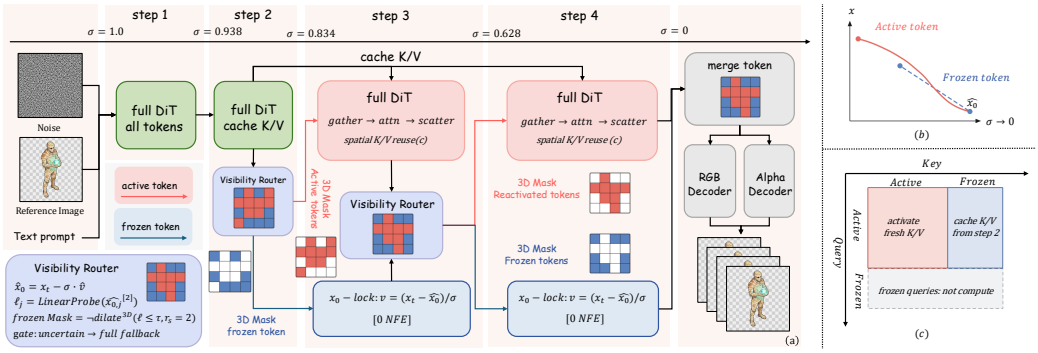


Figure 1: **Visibility-routed RGBA inference.** (a) Evaluations 1–2 process all tokens. After evaluation 2, the router uses $\hat{\mathbf{x}}_0^{[2]}$ to protect the predicted nontransparent support with a spatiotemporal margin and route the remaining tokens; uncertain samples retain dense inference. Evaluations 3–4 update active tokens, advance routed states with x_0-lock, and reactivate tokens reached by the evolving support before merging both sets for RGB and alpha decoding. (b) x_0-lock transports each routed state under the unchanged global noise schedule toward its stored endpoint $\mathbf{x}_{0,j}^{\star}$. (c) Cache-KV lets active queries attend to fresh active K/V and cached routed K/V while omitting routed queries, output projections, and feedforward updates.

2.3 Efficient Diffusion Inference

Efficient diffusion inference exploits redundancy across denoising steps, intermediate features, and tokens. Few-step distillation reduces denoising evaluations (Contributors, 2025; Wang et al., 2024), while cross-step reuse avoids repeated feature or attention computation (Liu et al., 2024a; Ma et al., 2024; Zou et al., 2024; Lyu et al., 2025). Token-level methods selectively update or merge tokens using previous-step outputs, feature similarity, or learned scores (Liu et al., 2025; Bolya et al., 2023; Bolya & Hoffman, 2023; Rao et al., 2021; 2022), but these criteria are not tied to RGBA visibility. Our router instead uses predicted final transparency as a task-specific token-selection criterion.

3 Method

Our framework addresses three challenges: constructing audited alpha supervision for synthesized RGB clips, incorporating alpha without expanding the DiT token sequence or adding an alpha-specific DiT branch, and estimating final visibility early enough to reduce later denoising computation. We address them with GameAlpha-2.4K, a merged RGB-A latent that preserves the pretrained I2V interface, and a conservative visibility router that routes tokens predicted to be transparent in the final output away from later DiT computation, while x_0-lock advances their latent states.

3.1 The GameAlpha-2.4K Dataset

Existing matting datasets provide limited coverage of diverse silhouettes, equipment, effects, and partial transparency found in game assets. A naive sequential pipeline samples prompts, synthesizes reference images and RGB videos, applies a single matte extractor, and filters only the final outputs. Even with category-balanced prompts, category-dependent rejection can bias the accepted set toward simpler concepts. Arbitrary backgrounds make foreground and alpha recovery underconstrained, while a single extractor propagates model-specific errors. We address these weaknesses through acceptance-aware sampling, matte-friendly synthesis, multiple alpha hypotheses, and stage-wise audits while failures remain repairable.

Balancing the dataset distribution. We use acceptance-aware sampling with compatibility constraints and adaptive weights over entity, equipment, action, style, and viewpoint (Appendix A.1).

Constraining foreground recovery before matting. Because a matte extractor cannot resolve foreground–background ambiguities already baked into RGB, we constrain generation before matting. A fixed high-capacity text-to-image model synthesizes a 1920×1920 reference image with a single frame-safe subject on a content-free background. A VLM then checks the sampled attributes,

framing, background contamination, and external cast shadows. Each rejection becomes a prompt repair instruction, and the reference is regenerated for up to three attempts. BiRefNet extracts an alpha matte from the accepted reference image to form an RGBA reference. The accepted reference RGB also conditions a fixed high-capacity image-to-video generator. A raw-video gate then rejects inconsistent subjects, incorrect motion or orientation, external shadows, scattered debris, and content-bearing backgrounds before matting.

Alpha recovery and staged audits. No single extractor performs consistently well across hard boundaries, fine structures, motion blur, holes, and partial opacity. We therefore generate three complete alpha hypotheses for each retained RGB clip using framewise BiRefNet, SAM3 video segmentation, and MatAnyone2 video matting. Since BiRefNet processes each frame independently, we add a temporal stabilizer that warps neighboring-frame mattes to the current frame using optical flow. This suppresses brief errors without averaging misaligned motion boundaries. A fixed multimodal evaluator compares sampled-frame composites for subject completeness, background leakage, soft boundaries, and temporal stability, then selects one complete sequence per clip; clip-level selection avoids discontinuities from switching extractors across frames. The selected alpha and generated RGB clip form the RGBA video. The VLM audits every selected clip for subject consistency, action, orientation, equipment, style, temporal stability, and foreground cleanliness. Human auditors independently review a prespecified 10% subset (Appendices A.1 and A.2).

Recurring failures and category imbalance update the next batch's compatibility rules, penalties, and sampling weights. Each sample includes an RGBA reference and video, the selected grayscale alpha sequence, structured image and video prompts, and an audit record.

3.2 End-to-End RGBA Generation

Preserving pretrained interfaces. Reference-conditioned RGBA generation must introduce alpha without disrupting the pretrained model's video token geometry, native reference conditioning, and motion prior. Alpha representation therefore determines DiT cost. TransPixeler doubles visual tokens from L to $2L$, expanding the full attention matrix from $(L_{\text{text}} + L)^2$ to $(L_{\text{text}} + 2L)^2$, while TransText enlarges the spatial canvas by concatenating RGB and replicated alpha. Concatenating RGB and alpha along the channel dimension requires new projection layers, which behave like training from scratch and reduce generation diversity when RGBA training data are limited. We therefore follow Wan-Alpha and use a merged RGB-A VAE that preserves Wan's native 16-channel latent. For an RGBA image or video $U = (U_{\text{rgb}}, U_\alpha)$, define

$$\Phi(U) = \mathcal{M}(\mathcal{E}(U_{\text{rgb}}), \mathcal{E}(\text{Rep}_3(U_\alpha))), \tag{1}$$

Here, $\mathcal{E}$ is the frozen Wan-VAE encoder, Rep_3 replicates grayscale alpha into three channels, and $\mathcal{M}$ is the learned RGB-A merge module. The target latent $\mathbf{x}_0 = \Phi(V_{\text{tar}})$ retains the pretrained geometry and normalization. Separate decoders recover RGB and alpha from the merged latent without extra DiT tokens or an alpha-specific DiT branch.

Reusing rather than rebuilding image conditioning. A compact alpha latent still requires reference control. Controllers such as SparseCtrl, I2V-Adapter, and VACE (Guo et al., 2023; 2024; Jiang et al., 2025) allow text-to-video backbones to accept image input, but they do not reduce the cost of an expanded alpha representation. Thus, we reuse Wan-I2V's native first-frame latent, CLIP-image, and umT5-text conditioning (Radford et al., 2021; Chung et al., 2023). Let $\mathbf{m}$ denote Wan-I2V's native four-channel first-frame mask, active only at the first latent frame. We form an RGBA conditioning video whose first frame is I_{ref} and whose subsequent frames are zero:

$$\begin{aligned} &I_{\text{cond}} = \text{Concat}_T(I_{\text{ref}}, 0, \ldots, 0),\ \mathbf{z}_{\text{cond}} = \Phi(I_{\text{cond}}), \\ &\mathbf{y} = \text{Concat}_C(\mathbf{m}, \mathbf{z}_{\text{cond}}) \in \mathbb{R}^{20 \times T_\ell \times H_\ell \times W_\ell}, \\ &\mathbf{u}_t = \text{Concat}_C(\mathbf{x}_t, \mathbf{y}) \in \mathbb{R}^{36 \times T_\ell \times H_\ell \times W_\ell}. \end{aligned} \tag{2}$$

The white-background reference composite and prompt produce $\mathbf{c}_{\text{img}}$ and $\mathbf{c}_{\text{text}}$ through the pretrained CLIP and umT5 pathways, respectively, and enter native decoupled cross-attention. This construction preserves Wan-I2V's 36-channel input and its native reference-conditioning interface without extra alpha tokens or an auxiliary ControlNet branch (Zhang et al., 2023).

Adapting the VAE and DiT. We adapt the RGB-A representation and video generator in two stages. Stage 1 learns the RGB-A latent contract using Wan-Alpha's VAE objective on the GameAlpha-2.4K training split and VideoMatte240K (Lin et al., 2021a). We freeze the pretrained encoder and base decoder, optimize only $\mathcal{M}$ and LoRA adapters (Hu et al., 2022) in the separate RGB and alpha decoders, and then freeze the complete RGB-A VAE. With the RGB-A representation fixed, Stage 2 adapts the pretrained RGB video generator to reference-conditioned RGBA generation by training rank-32 DoRA adapters (Liu et al., 2024b) on the GameAlpha training split while keeping the base DiT frozen. Given Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and timestep t with noise level σ_t, flow matching (Lipman et al., 2022) defines

$$\mathbf{x}_t = (1 - \sigma_t)\mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, \ \mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0, \ \hat{\mathbf{x}}_0^{(t)} = \mathbf{x}_t - \sigma_t \hat{\mathbf{v}}_t. \tag{3}$$

$$\hat{\mathbf{v}}_t = \hat{\mathbf{v}}_\theta(\mathbf{u}_t, \mathbf{c}_{\text{img}}, \mathbf{c}_{\text{text}}, t) \tag{4}$$

We retain the base scheduler's timestep-weighted objective

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}, t} \left[\omega(t) \ \text{MSE}\left(\hat{\mathbf{v}}_t, \mathbf{v}\right) \right]. \tag{5}$$

where $\omega(t)$ is the scheduler's timestep-dependent weight. The adapters are inserted in the DiT attention and feedforward layers.

Although decoded RGB does not affect a composite wherever $\alpha = 0$, each merged latent token jointly represents RGB and alpha, so downweighting it would also weaken zero-alpha supervision. The generator therefore retains the spatially uniform, scheduler-weighted objective in Eq. (5), while predicted visibility is used only by the inference router.

3.3 Visibility-Routed Inference

Predicting final transparency. Generic accelerators infer importance from attention, feature similarity, or update magnitude, whereas RGBA provides an analytic criterion. For RGB R, alpha α, and background B, let $C = \alpha R + (1 - \alpha)B$. For a prediction $(\hat{R}, \hat{\alpha})$ on the same background, define $\Delta R = \hat{R} - R$ and $\Delta\alpha = \hat{\alpha} - \alpha$. Then

$$\begin{aligned} \frac{\partial C_c}{\partial R_c} &= \alpha, \qquad c \in \{r, g, b\}, \\ \hat{C} - C &= \alpha \odot \Delta R + \Delta\alpha \odot (\hat{R} - B). \end{aligned} \tag{6}$$

Thus, RGB has no direct compositing contribution where final alpha is zero. The criterion is analytic at the final output, but routing must predict which tokens will be transparent during denoising.

We do not use the dataset target alpha as a routing label because it may not align with the generated motion. Likewise, simply repeating the first-frame alpha is also insufficient because it cannot anticipate future motion. We instead use the clean estimate $\hat{\mathbf{x}}_0^{[k]} = \mathbf{x}_k - \sigma_k \hat{\mathbf{v}}_k$ at DiT evaluation k, following Eq. (3). Square brackets index evaluations rather than diffusion time. A logistic-regression probe, fitted offline on dense trajectories from the training clips and frozen for sparse inference, maps each token's $2{\times}2{\times}16$ latent patch to a 64-dimensional feature and transparency logit ℓ_j. Dense final generations provide self-labels: token j is effectively transparent if the maximum decoded alpha over its spatiotemporal region Ω_j is at most $\eta = 0.05$, with larger ℓ_j indicating greater confidence in this event. These self-labels target fidelity to dense generation at lower cost, not independent alpha accuracy; Appendix C.2 analyzes early prediction of final transparency. Routing time trades prediction risk against the number of remaining sparse evaluations. The training-split risk–yield analysis in Sec. 4.5 supports routing after evaluation 2, so we set $k_{\text{route}} = 2$ and retain the final two evaluations for sparse execution under the unchanged four-step sampling schedule.

A safety margin and dense fallback. Because foreground support may expand after evaluation 2, routing uses a conservative spatiotemporal margin. With $\tau = 0$, tokens satisfying $\ell_j > \tau$ are predicted transparent; we dilate the complementary nontransparent support by $r_s = 2$, extend it by $r_t = 1$ along latent time, and always keep the first latent frame active. Tokens outside the protected support form the candidate routed set. The same training-split risk–yield analysis in Sec. 4.5 supports $r_s = 2$, while $r_t = 1$ provides an adjacent-time guard. The evolution of decoded alpha and the spatial distribution of support mismatches provide complementary diagnostics for the safety margin and subsequent mask revision (Appendices C.3 and C.4). Before constructing sparse states or caches, the sample remains on the dense path if fewer than 10% of tokens are candidates or more than 15% are uncertain, where uncertainty is defined by $|\ell_j - \tau| < \delta$ with $\delta = 2$.

Skipping with x_0-lock. Routing determines which later token updates may be omitted, but not how the latent states should evolve. Holding a state fixed leaves it at an obsolete noise level. Assigning different timesteps to individual tokens creates an asynchronous schedule that conflicts with the pretrained DiT's global time conditioning. For each token j routed after $k_{\text{route}} = 2$, we store the model's clean endpoint $\mathbf{x}^{\star}_{0,j} = \hat{\mathbf{x}}^{[2]}_{0,j}$. Evaluation 2 still uses the ordinary dense scheduler update. At each later evaluation $k > k_{\text{route}}$, the routed token bypasses the DiT and follows

$$\begin{aligned} \mathbf{v}^{\text{lock}}_{k,j} &= \frac{\mathbf{x}^{\text{lock}}_{k,j} - \mathbf{x}^{\star}_{0,j}}{\sigma_k}, \\ \mathbf{x}^{\text{lock}}_{k+1,j} &= \mathbf{x}^{\text{lock}}_{k,j} + (\sigma_{k+1} - \sigma_k)\mathbf{v}^{\text{lock}}_{k,j} \\ &= \mathbf{x}^{\star}_{0,j} + \frac{\sigma_{k+1}}{\sigma_k}\left(\mathbf{x}^{\text{lock}}_{k,j} - \mathbf{x}^{\star}_{0,j}\right). \end{aligned} \tag{7}$$

Because $(\mathbf{x}^{\text{lock}}_{k,j} - \mathbf{x}^{\star}_{0,j})/\sigma_k$ remains constant, Eq. (7) exactly transports each routed state along the straight flow-matching interpolant defined by its stored endpoint under the unchanged global noise schedule. Only the evaluation 2 estimate is approximate. Using token-specific endpoints also avoids discontinuities from a shared transparent latent.

Local reactivation and routed-token context. Although the evaluation-2 safety margin protects likely foreground expansion, the predicted support may still change during the sparse suffix. After the first sparse evaluation, we update the protected nontransparent support from the fresh clean estimates of active tokens using the same spatial and temporal margins. Any routed token reached by this updated support is reactivated for the final evaluation. Because x_0-lock has advanced its state, the token re-enters the DiT at the current global noise level. For tokens that remain routed, attention context is the remaining concern: even a token with zero final alpha may influence active queries despite having no direct RGB contribution in Eq. (6). Our primary Cache-KV implementation therefore stores the per-layer K/V of routed tokens at evaluation 2, the last dense evaluation. During later sparse evaluations, active tokens compute fresh Q/K/V and full attention and feedforward updates, attending to fresh active and cached routed K/V; routed tokens reuse cached K/V, omit queries, output projections, and feedforward computation, and advance only through Eq. (7). When a token is reactivated, its cached rows are replaced with fresh K/V after re-entry. Drop-KV removes routed context for greater speed. Both variants retain four evaluations without changing the backbone parameters or denoising schedule. Algorithm 1 in Appendix B.2 summarizes the sampling procedure.

4 Experiment

4.1 Setup

Protocol. After splitting GameAlpha-2.4K into 2,156 training and 239 validation clips, we train the Stage 1 RGB-A VAE on VideoMatte240K and the training clips, and the Stage 2 DiT only on the latter. The transparency probe is fitted on dense training trajectories and frozen. We select $(k_{\text{route}}, r_s) = (2, 2)$ using dense trajectories from the GameAlpha training split and fix them before validation; unless otherwise stated, GameAlpha metrics use the validation split at a fixed seed. We additionally evaluate dense and routed inference across three paired seeds and report the mean and standard deviation of FVD across seeds (Appendix C.5). For the external VideoMatte240K evaluation, we use a separately trained RGB-A VAE that excludes this dataset from training. We transfer the routing configuration unchanged to VideoMatte240K, without tuning routing parameters or selecting checkpoints using its results. It also performs best among the routing settings evaluated there, supporting cross-domain transfer (Appendix C.7). Inference follows the four-evaluation schedule and dense fallback in Sec. 3.3.

Metrics. FVD (Unterthiner et al., 2019; Skorokhodov et al., 2021; Yu et al., 2022; Carreira & Zisserman, 2017) uses I3D features from checkerboard composites. DINO-ID averages DINOv2 (Oquab et al., 2024) similarity between generated frames and the reference, a LAION predictor scores frame aesthetics, and motion smoothness follows VBench (Huang et al., 2024). MAD (Lin et al., 2021a; Yang et al., 2026) measures agreement with the dataset alpha sequence, reported in 10^{-3}, and is sensitive to differences in generated motion (Appendix B.4). Flow Difference adapts TransPixeler: it compares consecutive-frame Farneback flows (Farnebäck, 2003) of

Table 1: Comparison with fine-tuned two-stage generate-then-matte systems on the GameAlpha-2.4K validation set. Time is mean end-to-end seconds per clip; MAD is in 10^{-3}. **Bold** marks the best value in each column; ties at displayed precision are also bolded.

Generator	Matte	FVD↓	DINO-ID↑	Aes.↑	Motion Sm.↑	Flow-Diff↓	MAD↓	Time↓
Wan2.1-I2V (4-step)	BiRefNet	224.3	0.848	5.396	0.984	0.598	160.0	128.4
	MatAnyone2	225.8	0.846	**5.402**	0.985	0.586	171.3	149.9
	SAM3	221.3	**0.849**	5.369	0.985	0.611	159.6	126.7
	UniVidX	276.6	0.695	5.116	0.986	**0.548**	194.9	4212.9
CogVideoX1.5-I2V	BiRefNet	318.8	0.848	5.223	0.986	0.609	**92.9**	664.5
	MatAnyone2	320.2	0.846	5.234	0.985	0.620	93.9	688.0
	SAM3	316.2	0.845	5.202	0.985	0.688	94.0	655.7
	UniVidX	330.7	0.733	5.011	0.987	0.595	138.6	4749.0
HunyuanVideo-I2V	BiRefNet	418.6	0.790	5.142	0.985	0.688	261.8	2281.1
	MatAnyone2	399.6	0.794	5.121	0.985	0.660	236.2	2291.7
	SAM3	391.8	0.804	5.124	0.986	0.658	242.9	2272.4
	UniVidX	470.6	0.515	4.725	0.986	0.632	237.8	6377.5
Ours (visibility router)		**174.4**	0.840	5.314	**0.988**	**0.548**	108.1	**104.8**

the white-background RGB composite and alpha, averages endpoint distance within a dilated foreground, and normalizes this distance by the sum of the mean magnitudes of both fields. Both tables report mean end-to-end wall-clock time per clip, including matting for two-stage systems. Dense and routed variants in Table 2 use the same checkpoint and matched inference settings at batch size 1. Timing covers condition encoding, denoising, RGB/alpha decoding, video encoding, and file writing, including all routing overhead and excluding model loading (Appendix B.3).

4.2 Dataset Analysis

GameAlpha-2.4K provides structured coverage across seven entity categories, five action categories, and 23 visual styles. Every permitted entity–action combination is represented, with the taxonomy and observed coverage detailed in Appendices A.1 and A.3. Each GameAlpha-2.4K clip contains 97 frames at a native resolution of 960×960. Prior corpora focus on humans/VFX, generic objects, game effects, or glyphs, while LayerDiffuse is image-only; GameAlpha instead targets reusable game assets, contributing domain coverage rather than raw scale (Lin et al., 2021a). Appendix A.6 compares dataset coverage. Intermediate-opacity pixels ($2/255 < \alpha < 253/255$) account for 2.2% of all frame pixels, including fully transparent background regions in the denominator. This measures pixel area rather than the prevalence of clips containing soft transparency: fine boundaries, hair, glow, and motion blur can occupy localized regions (Appendix A.4). The broader Stage 1 training mixture complements this distribution (Sec. 3.2).

Implementation. We build on Wan2.1-I2V-14B with the RGB-A VAE in Sec. 3.2 and four-step LightX2V distillation (Contributors, 2025). We optimize rank-32 DoRA modules in the DiT attention projections (q, k, v, o) and feedforward layers for 10 epochs with AdamW (Loshchilov et al., 2017) at 1.4×10^{-4} using distributed bfloat16 training. Training completes in less than a day on a 32-GPU node. We generate 97 frames at 624×640.

4.3 Comparison with Two-Stage Pipelines

We evaluate complete reference-conditioned RGBA generation systems against fine-tuned generate-then-matte pipelines in Table 1. The two-stage systems pair three I2V generators—Wan2.1-I2V-14B (Wan et al., 2025), CogVideoX1.5-I2V (Yang et al., 2024), and HunyuanVideo-I2V (Kong et al., 2024)—with four alpha extractors: BiRefNet (Zheng et al., 2024), MatAnyone2 (Yang et al., 2026), SAM3 (Carion et al., 2025), and UniVidX (Chen et al., 2026a). All three generators are fine-tuned on the GameAlpha training split; Wan uses the same DiT adaptation recipe and budget as ours (Appendices B.1 and B.3). Its four-step system shares our backbone and step budget. All video generators receive the same 624×640 white-composited reference and output 97 frames; ours additionally uses the RGBA latent anchor in Sec. 3.2, while MatAnyone2 and SAM3 are initialized with the same reference alpha. All systems are evaluated on the same validation videos and dataset alpha sequences.

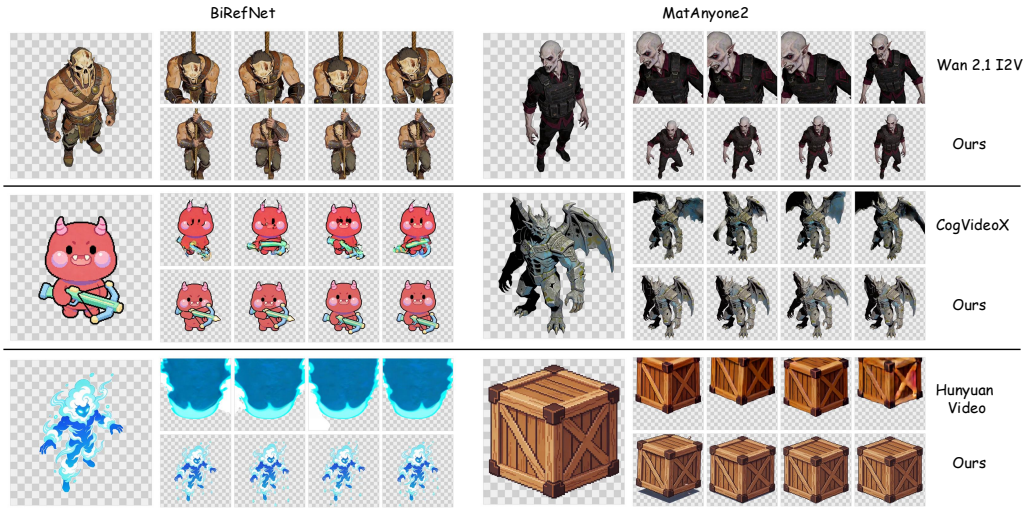


Figure 2: **Qualitative comparison with two-stage baselines.** We pair three RGB generators with BiRefNet or MatAnyone2. Each group shows the RGBA reference (left) and four uniformly sampled frames from the two-stage baseline (top) and ours (bottom). Checkerboards denote transparency; prompts are provided in Appendix D.3.

Table 2: Single-seed routing ablation on GameAlpha-2.4K. Time is end-to-end seconds; MAD is 10^{-3}. Initial routing averages 35%; RAS, Planner, Union2D, and Random match realized skips. **Bold**/underlined: best/second-best sparse quality.

Variant	FVD↓	DINO-ID↑	Aes.↑	Motion Sm.↑	Flow Diff.↓	MAD↓	Time↓	Speedup↑
Ours (dense)	172.3	0.840	5.317	0.988	0.550	108.2	125.9	1.00×
Ours (Cache-KV)	**174.4**	**0.840**	**5.314**	**0.988**	0.548	108.1	104.8	1.20×
Drop-KV	177.2	**0.840**	5.304	0.987	0.549	108.9	96.0	1.31×
No reactivation	177.2	0.838	5.293	0.987	0.551	110.5	104.1	1.21×
RAS proxy	819.3	0.823	5.196	0.985	**0.462**	109.5	105.1	1.20×
Planner (static)	181.6	0.838	5.290	0.986	0.547	108.5	105.2	1.20×
Union2D (static)	184.2	0.836	5.273	0.987	0.540	108.4	105.2	1.20×
Random	297.1	0.820	5.193	0.984	0.532	**107.7**	105.2	1.20×

Table 1 shows that our full model achieves the lowest FVD (174.4) and runtime (104.8s). Alpha metrics must be interpreted with content quality: our MAD is below every Wan and Hunyuan-Video pipeline, although three CogVideoX combinations obtain lower MAD with much worse FVD. Our model also achieves the best motion smoothness and keeps DINO-ID within 0.009 of the best pipeline. Figure 2 illustrates qualitative differences in prompt following, foreground preservation, and checkerboard composites in the displayed examples.

To assess dependence on supplied reference alpha, we evaluate a variant that estimates the first-frame alpha solely from the RGB reference. For each of the 239 validation inputs, both conditions use the same checkpoint, RGB input, text, and initial noise, and run on the same GPU with routing disabled. With only RGB and text supplied, the system obtains FVD 172.79, compared with 171.56 using dataset-provided reference alpha; mean end-to-end runtime increases from 125.34 to 126.55 seconds, including alpha estimation (Appendix C.10).

Beyond GameAlpha, our routed model achieves FVD 125.7 and MAD 105.2×10^{-3} on 484 out-of-domain VideoMatte240K clips (Appendices C.7 and D.5). Representative frames from generated videos on additional out-of-domain inputs exhibit fine smoke structures and gradual opacity transitions when the same foreground and alpha are composited onto checkerboard, white, and black backgrounds (Appendix D.4). Together, the quantitative cross-domain evaluation and qualitative out-of-domain examples support transfer beyond GameAlpha following DiT adaptation on its compact training split. Blinded human assessments additionally evaluate RGB–alpha alignment in generated videos using independent ratings from ten professional annotators (Appendix C.9).

4.4 Ablation Studies

Our dense model serves as a direct RGBA-I2V control, sharing the RGB-A representation, checkpoint, and reference conditioning with the routed model. Their comparison isolates the effect of visibility routing within the same generator.

Table 2 also compares visibility routing with a RAS-style update-magnitude proxy (Liu et al., 2025), two static spatial policies, and random selection (Appendix B.3). Our primary router uses Cache-KV; Drop-KV retains visibility-based selection and reactivation but discards routed K/V context. The no-reactivation variant otherwise matches the primary router but fixes the evaluation-2 routed set, isolating the value of revising early decisions. *(i) The signal matters.* Random and RAS selection can freeze visible content, degrading FVD to 297.1 and 819.3, respectively. A small update at the routing evaluation does not imply final transparency, so update-magnitude saliency may freeze still-evolving content. *(ii) Adaptivity matters.* Static spatial policies reach FVD 181.6/184.2 but trail the router in DINO-ID, aesthetics, and motion smoothness, showing that fixed support cannot track evolving motion. *(iii) Visibility routing remains close to dense inference.* Cache-KV changes FVD from 172.3 to 174.4 while reducing end-to-end runtime from 125.9 to 104.8 s (1.20×) and matching dense DINO-ID and motion smoothness. Drop-KV takes 96.0 s (1.31×) with FVD 177.2, indicating a modest benefit from retaining routed K/V context. MAD remains near dense. Flow Difference diagnoses RGB–alpha motion alignment: RAS scores 0.462 despite FVD 819.3, so it must be interpreted with content quality. *(iv) Matched reactivation.* The three-seed paired evaluation corroborates near-dense FVD at a 1.20× end-to-end speedup and shows lower mean FVD with reactivation at a small additional runtime cost (Appendix C.5). Paired qualitative examples from the matched reactivation evaluation are provided in Appendix D.2. Per-clip routing fractions, dense-fallback frequencies, and reactivation statistics are reported in Appendix C.6.

4.5 Routing Time and Safety Margin

Because final transparency is known only after sampling, routing must balance computation savings against the risk of omitting updates for tokens that later become visible. On GameAlpha training trajectories, Appendix Figure 9 plots P90 final-visible-token miss against median full-trajectory token-evaluation saving across routing times and spatial margins. At $r_s = 2$, delaying routing from evaluation 1 to 2 reduces P90 miss from 0.824% to 0.603%, while leaving fewer token updates to skip. We select evaluation 2 because it lowers tail risk while retaining two sparse evaluations.

The full GameAlpha validation set contains clips with varying transparent-region sizes (Appendix A.4), and the visibility router adapts the routed set to each clip's predicted transparency. Foreground-dense clips retain more tokens for full updates; clips with fewer than 10% eligible tokens after the safety margin remain on the dense path before sparse states are constructed. Across all 239 validation clips, the mean initial routed fraction after evaluation 2 is 35%; reactivation can reduce subsequent skips. Under this adaptive policy, mean end-to-end runtime over the entire validation set decreases from 125.9 to 104.8 s, yielding a 1.20× speedup over our dense model computed as the ratio of these means (Table 2). Appendices C.1 and C.6 provide full-trajectory token-evaluation saving definitions and per-clip routing and fallback statistics.

5 Conclusions, Limitations and Future Work

Alpha determines compositing and where further RGB refinement affects the result. GameAlpha-2.4K supplies game-asset data, while a merged RGB-A latent lets one image-conditioned DiT generate color and alpha jointly. This native alpha enables the visibility router to identify final-transparent tokens; x_0-lock advances them on the unchanged flow schedule, while conservative fallback protects uncertain clips. Experiments show lower FVD than the evaluated fine-tuned generate-then-matte systems and measured quality close to dense at a 1.20× end-to-end speedup over our dense model. The binary router targets tokens predicted effectively transparent ($\alpha \leq 0.05$), limiting acceleration on foreground-dense clips. Routing errors also remain nonzero. GameAlpha-2.4K uses synthesized videos and extractor-selected alpha, with intermediate-opacity pixels accounting for 2.2% of all frame pixels (Sec. 4.2). Future work will study computation allocation by compositing error, artist- or renderer-derived RGBA supervision, and broader evaluations of cross-domain transfer, compositing on diverse backgrounds, and compatibility with additional video backbones.

AI USE STATEMENT

Generative AI systems were used as components of the dataset-construction pipeline: an LLM expanded accepted structured tuples into image and video prompts and audit metadata; fixed high-capacity text-to-image and image-to-video systems synthesized candidate references and videos; and vision-language or multimodal systems supported quality gating and clip-level candidate selection. Human auditors reviewed a prespecified subset, as detailed in Appendix A.1. A generative AI assistant also helped polish the language of the manuscript. The authors reviewed the AI-assisted work and take responsibility for the final text, claims, and artifacts.

ETHICS STATEMENT

The human evaluation involved ten professional annotators in a blinded, within-subject protocol. Only anonymized participant labels and aggregate ratings are reported. The proposed dataset, code, and model release will preserve upstream artifact licenses and will proceed only after verifying that the terms of the generation and screening services permit redistribution of generated outputs. Additional study details and third-party licensing considerations appear in Appendices C.9 and F.

REPRODUCIBILITY STATEMENT

The main paper specifies the model, training objective, routing mechanism, and evaluation protocol. The supplementary material documents dataset construction and gates (Appendix A), training and inference settings, router pseudocode, baselines, and metric definitions (Appendix B), additional analyses and sample sizes (Appendix C), and the full qualitative prompts (Appendix D.3). Data splits and exceptions are summarized in Appendix E. The dataset, implementation, and checkpoint are not public at submission; their planned release status is stated in Appendix F.

REFERENCES

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. *CVPR Workshop on Efficient Deep Learning for Computer Vision*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. URL https://api.semanticscholar.org/CorpusID:206596127.

Houyuan Chen, Hong Li, Xianghao Kong, Tianrui Zhu, Shaocong Xu, Weiqing Xiao, Yuwei Guo, Chongjie Ye, Lvmin Zhang, Hao Zhao, and Anyi Rao. Unividx: A unified multimodal framework for versatile video generation via diffusion priors. *ACM Transactions on Graphics*, 45(4), July 2026a. doi: 10.1145/3811304. URL https://doi.org/10.1145/3811304.

Xuewei Chen, Zhimin Chen, and Yiren Song. Transanimate: Taming layer diffusion to generate rgba video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6083–6092, 2026b.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining, 2023. URL https://arxiv.org/abs/2304.09151.

LightX2V Contributors. Lightx2v: Light video generation inference framework. https://github.com/ModelTC/lightx2v, 2025.

Haotian Dong, Wenjing Wang, Chen Li, Jing Lyu, and Di Lin. Video generation with stable transparency via shiftable rgb-a distribution learner, 2025. URL https://arxiv.org/abs/2509.24979.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conference on Image Analysis*, 2003. URL https://api.semanticscholar.org/CorpusID:15601477.

Xun Guo, Mingwu Zheng, Liang Hou, Yuan Gao, Yufan Deng, Pengfei Wan, Di Zhang, Yufan Liu, Weiming Hu, Zhengjun Zha, et al. I2v-adapter: A general image-to-video adapter for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–12, 2024.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models, 2023.

Edward J Hu, Yalu Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Sihui Ji, Hao Luo, Xi Chen, Yuanpeng Tu, Yiyang Wang, and Hengshuang Zhao. Layerflow : A unified model for layer-aware video generation. *arXiv preprint arXiv:2506.04228*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

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.

Jiachen Li, Vidit Goel, Marianna Ohanyan, Shant Navasardyan, Yunchao Wei, and Humphrey Shi. Vmformer: End-to-end video matting with transformer. *arXiv preprint*, 2022.

Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. *arXiv: 2306.05399*, 2023.

Menghao Li, Zhenghao Zhang, Junchao Liao, Long Qin, and Weizhi Wang. Transvdm: Motion-constrained video diffusion model for transparent video synthesis. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

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.

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*, pp. 8762–8771, 2021a.

Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. *arXiv preprint arXiv:2411.19108*, 2024a.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024b.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers, 2025. URL https://arxiv.org/abs/2502.10389.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Zhengyao Lyu, Chenyang Si, Junhao Song, Zhenyu Yang, Yu Qiao, Ziwei Liu, and Kwan-Yee K Wong. Fastercache: Training-free video diffusion model acceleration with high quality. In *International Conference on Learning Representations*, volume 2025, pp. 33132–33156, 2025.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024.

Yifan Pu, Yiming Zhao, Zhicong Tang, Ruihong Yin, Haoxing Ye, Yuhui Yuan, Dong Chen, Jianmin Bao, Sirui Zhang, Yanbin Wang, et al. Art: Anonymous region transformer for variable multi-layer transparent image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7952–7962, 2025.

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

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yongming Rao, Zuyan Liu, Wenliang Zhao, Jie Zhou, and Jiwen Lu. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *arXiv preprint arXiv:2207.01580*, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *arXiv preprint arXiv:2112.14683*, 2021.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLR 2019 Workshop on Deep Generative Models for Highly Structured Data*, 2019.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang,

Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Xiaogang Wang, and Hongsheng Li. Phased consistency models, 2024. URL https://arxiv.org/abs/2405.18407.

Luozhou Wang, Yijun Li, Zhifei Chen, Jui-Hsien Wang, Zhifei Zhang, He Zhang, Zhe Lin, and Ying-Cong Chen. Transpixeler: Advancing text-to-video generation with transparency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18229–18239, 2025.

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

Peiqing Yang, Shangchen Zhou, Kai Hao, and Qingyi Tao. MatAnyone 2: Scaling video matting via a learned quality evaluator. In *CVPR*, 2026.

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.

Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Czsdv-S4-w9.

Fei Zhang, Zijian Zhou, Bohao Tang, Sen He, Hang Li, Zhe Wang, Soubhik Sanyal, Pengfei Liu, Viktar Atliha, Tao Xiang, et al. Transtext: Alpha-as-rgb representation for transparent text animation. *arXiv preprint arXiv:2603.17944*, 2026.

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *arXiv preprint arXiv:2402.17113*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. *arXiv preprint arXiv:2410.05317*, 2024.

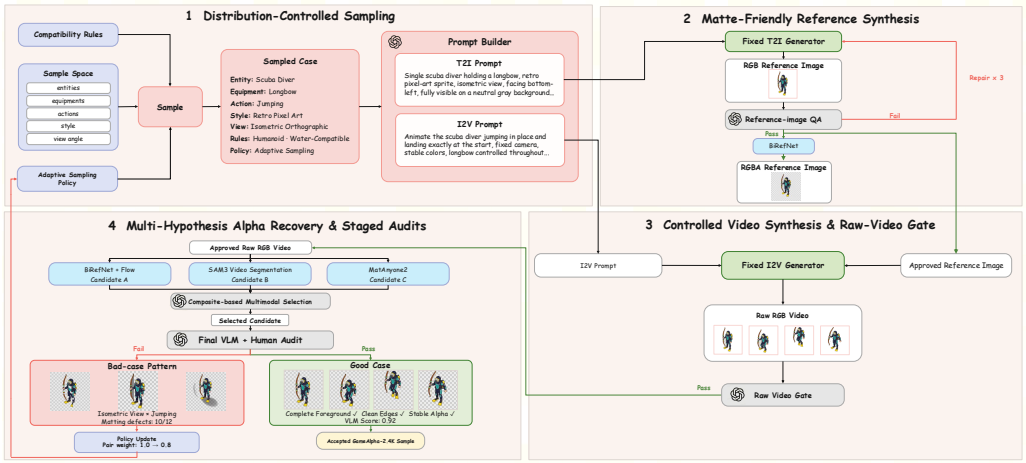


Figure 3: **The GameAlpha-2.4K construction pipeline.** Distribution-controlled prompt sampling (Stage 1) feeds matte-friendly reference synthesis with a repair loop (Stage 2). The approved RGB reference drives video synthesis and a raw-video gate (Stage 3), while a parallel branch forms the RGBA reference. Retained clips receive three alpha hypotheses for composite-based selection and final VLM-and-human audit (Stage 4). Recurring failures and category imbalance update the next batch's sampling rules, penalties, and weights (red feedback edge).

Supplementary Material

This supplementary material provides: (**A**) construction details, design rationale, and statistics of the GameAlpha-2.4K dataset; (**B**) implementation and evaluation-protocol details; (**C**) additional experiments, analyses of the visibility router, and human evaluation; (**D**) additional qualitative results and the prompts used by the qualitative figures; (**E**) reproducibility notes; and (**F**) licensing and planned release status.

A The GameAlpha-2.4K Dataset

A.1 Construction Pipeline

Figure 3 summarizes the four stages of the pipeline; the Sec. 3.1 describes each gate in one paragraph, and this section documents the design rationale and the operational details that did not fit in the main text.

Stage 1: distribution-controlled sampling. Each sample starts from a tuple over entity (7 classes), equipment (7 classes), action (5 classes), style, and viewpoint. Hard rules remove invalid combinations, and soft penalties with category weights suppress unlikely tuples while filling underrepresented combinations. An LLM expands each accepted tuple into a separate image prompt and video prompt, color metadata, and a structured description that later audits check against. Sampling weights are updated from the *accepted* distribution and the recorded failures rather than from the proposed distribution: acceptance rates differ across categories, so balancing proposals alone still lets the accepted set drift toward easy concepts. Table 3 lists the taxonomy. Hard rules encode physical or perspective incompatibilities (for example, isometric 2.5D rendering excludes strict top-down and pure side-profile viewpoints, and object or vehicle entities exclude character-portrait viewpoints), while soft penalties down-weight unlikely pairings such as minimalist geometric styles with semantically strong actions.

Stage 2: matte-friendly reference synthesis. An extractor cannot undo foreground–background ambiguity that is already baked into an RGB image, so the pipeline constrains generation *before* matting: a fixed high-capacity text-to-image generator renders a 1920×1920 reference with a single frame-safe subject on a content-free background, and a VLM gate checks the sampled attributes,

Table 3: Sampling taxonomy used to construct GameAlpha-2.4K. Category counts are shown in parentheses. Candidate tuples span all five dimensions; invalid combinations are filtered by hard rules, while category weights and soft penalties shape the proposal distribution (Appendix A.1).

Dimension	Classes
Entity (7)	Character/NPC; Monster/Creature; Wildlife/Animal; Vehicle/Transport; Environment Prop; Modern Tech; Food/Plant
Equipment (7)	Unarmed; Melee Weapon; Ranged Weapon; Armor/Apparel; Consumable; Tool/Gadget; Magic Relic
Action (5)	Locomotion; Combat Action; State/Reaction; NPC Daily; Physics/VFX
Style (23)	Core mainstream; genre-specific; long-tail artistic. The accepted clips cover 23 distinct styles.
Viewpoint (7)	Mainstream 2D; depth/strategy; character/UI. Seven named viewpoints are included.

framing, background contamination, and external cast shadows. Each rejection is converted into a prompt-repair instruction and the reference is regenerated for up to three attempts before the tuple is returned to the sampler as a recorded failure. BiRefNet extracts the extractor-derived reference alpha of the accepted image to form the RGBA reference.

Stage 3: controlled video synthesis and the raw-video gate. The accepted reference RGB conditions a fixed high-capacity image-to-video generator. A raw-video gate rejects inconsistent subjects, incorrect motion or orientation, external shadows, scattered debris, and content-bearing backgrounds *before* the matting stage, so defective clips never consume the three-extractor budget.

Stage 4: multi-hypothesis alpha recovery and staged audits. No single extractor is best across hard boundaries, fine structures, motion blur, holes, and partial opacity, so each retained clip receives three complete alpha hypotheses: framewise BiRefNet, SAM3 video segmentation, and MatAnyone2 video matting. Only the framewise BiRefNet candidate is stabilized by warping neighboring mattes into the current frame with optical flow and taking a temporal median, which suppresses brief errors without averaging misaligned motion boundaries; the two video models are left untouched. A fixed multimodal evaluator compares white-background composites of sampled frames for subject completeness, background leakage, soft boundaries, and temporal stability, and selects *one complete sequence per clip* — clip-level selection avoids the discontinuities that per-frame extractor switching would introduce. Across the accepted corpus, the selected sequence comes from BiRefNet for 81% of clips, SAM3 for 13%, and MatAnyone2 for 6%. The VLM audits every selected clip for subject consistency, action, orientation, equipment, style, temporal stability, and foreground cleanliness. Human auditors independently review a prespecified one-tenth subset as a sampled quality-control check; this human review is not presented as a full-corpus gate.

A.2 Gate Statistics and Audit Records

Figure 4 reports the acceptance funnel of the accepted batch. Two properties are worth noting. First, rejection is front-loaded and cheap: the reference gate and its prompt-repair loop act before video synthesis, and the matte-stage quality gate acts before any clip reaches the final audits, so expensive stages never process clips that earlier stages could already reject. Second, the final frame-safety review trades raw volume for the frame-safe property that reference-conditioned training relies on. Every accepted clip passes the VLM audit. Human auditors additionally review a prespecified one-tenth subset for subject identity, action alignment, viewpoint, equipment, and style; this sampled audit feeds the sampling-policy iteration of Appendix A.1 but is not a full-corpus human gate.

A.3 Category Coverage

Figures 5 and 6 show the realized category coverage. The support-completeness property is the operative check for a compositional taxonomy: all empty entity–action cells are those excluded by hard compatibility rules (vehicles and props do not perform daily-life or combat actions), not collapsed categories, and the smallest populated classes retain $\approx 5\%$ of the corpus. We report

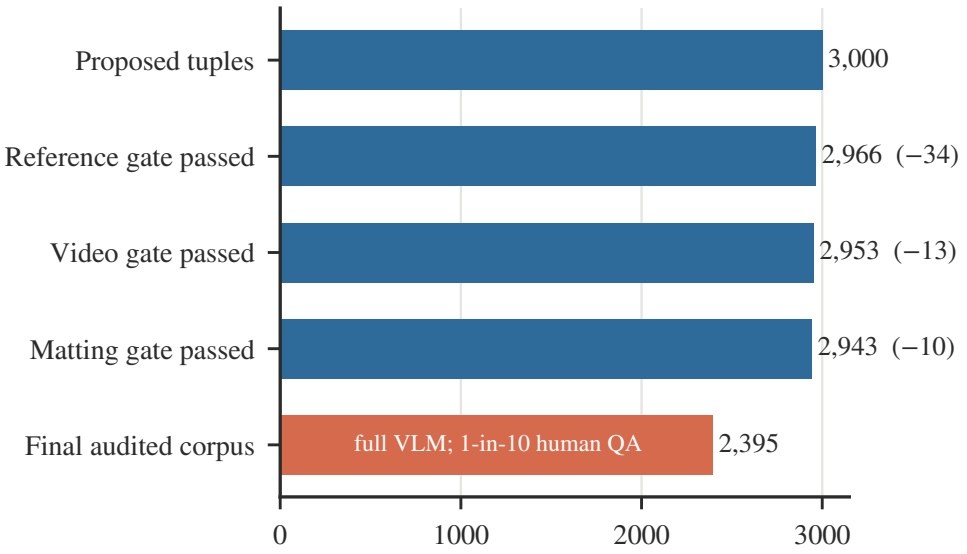


Figure 4: **Acceptance funnel of GameAlpha-2.4K.** From 3,000 proposed tuples, 34 are removed at the reference stage (25 rejected by the reference gate after up to three prompt-repair attempts, 9 synthesis-service failures), 13 at the video stage (synthesis-service and content screening), 10 by the matte-stage quality gate (per-frame extraction-failure rate above 25%). For 19 of the resulting 2,943 unique tasks, production retained one additional regenerated clip, giving 2,962 clip artifacts at the final frame-safety review; removing 567 edge-touching artifacts yields 2,395 accepted clips (2,156 train / 239 validation).

observed coverage only: proposal-side tuple distributions were not persisted with this batch, so no quantitative before/after reweighting comparison is claimed.

A.4 Transparency Statistics

Figure 7 summarizes the transparency structure of the dataset and connects it to two design choices in Sec. 3.2 and Sec. 3.3. The per-pixel distribution in panel (a) is the domain fingerprint of RGBA game assets: probability mass concentrates at the two ends, and the soft band — hair, glow, motion blur, semi-transparent effects — is narrow (2.2% of pixels) yet is exactly where compositing quality is judged. This sparsity of intermediate opacity is why Stage 1 (Sec. 3.2) trains the RGB-A VAE on a broader mixture rather than on GameAlpha-2.4K alone. Panel (b) contextualizes the initial routed fraction: on average 62.5% of tokens are fully transparent under the conservative pooling criterion, so the mean initial routed fraction of 35% after evaluation 2 lies below the fully transparent token fraction of nearly all validation videos (and the first latent frame is never routed, which lowers the implementation-reachable ceiling to 59.9% on average). Panel (c) quantifies why adaptivity matters: transparent sets barely move between adjacent latent frames but drift substantially away from any static first-frame plan, which is precisely the failure mode of the static baselines in the routing ablation of Table 2.

A.5 Sample Gallery

Figure 8 shows five samples spanning distinct render treatments (isometric 2.5D, cel-shaded, top-down pixel art, neon-accented flat 2D, painterly dark fantasy). Two properties recur. First, mattes remain clean across the clip at exactly the structures where post-hoc matting degrades — fur tufts, glow accents, thin props. Second, each motion is prompted to return to its starting pose within the frame-safe canvas, making the clip suitable for looped compositing workflows. For out-of-domain generated RGBA examples recomposited onto checkerboard, white, and black backgrounds, see Appendix D.4.

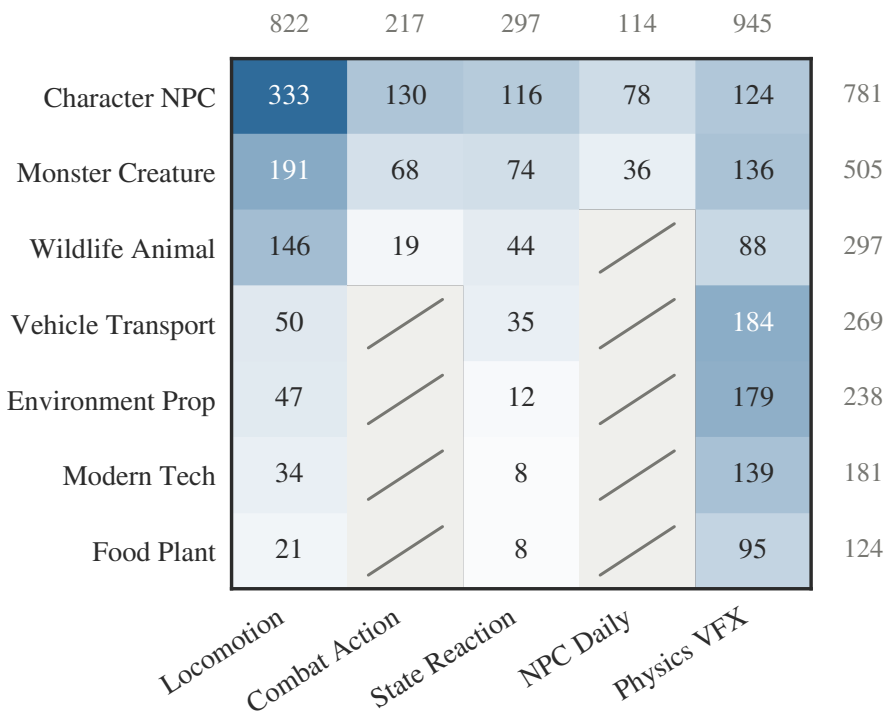


Figure 5: **Entity×action coverage** of GameAlpha-2.4K. Hatched cells are structurally forbidden combinations under the body-type compatibility rules (9 of 35); every supported cell is populated (minimum count 8). The smallest entity class holds 5.2% of clips and the smallest action class 4.8%.

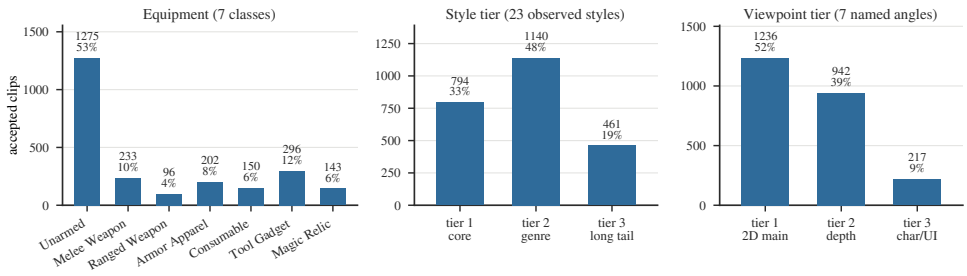
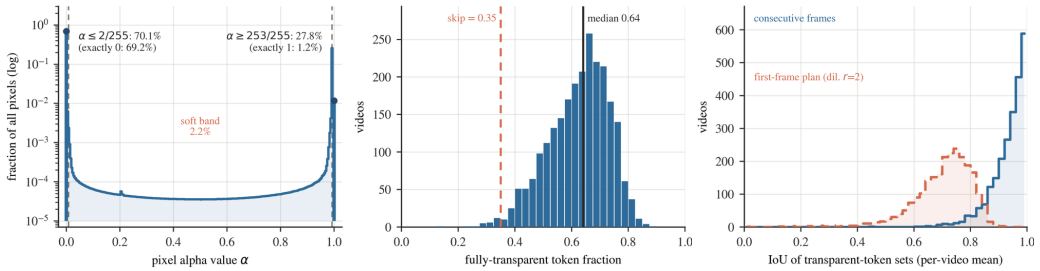


Figure 6: **Marginal distributions** over equipment, style tier, and viewpoint tier. Closed-loop reweighting keeps every category populated but not uniform: acceptance rates differ across categories, and sampling mixes a mainstream and an exploration mode, so observed shares deviate from proposal weights.

A.6 Comparison with Existing Dataset

Table 4 compares reported scale, domain, and availability of RGBA training datasets. GameAlpha-2.4K contains 2,395 clips with 97 frames each, totaling 232,315 frames at the 960×960 native resolution; "2.4K" is the rounded dataset name. The datasets differ in domain: green-screen human footage (VideoMatte240K (Lin et al., 2021a)), human/VFX clips (Wan-Alpha), generic objects (TransVDM), game effects (TransAnimate), and glyphs (TransText) do not cover reusable game-asset animation with equipment, stylization, and effect-like foregrounds; LayerDiffuse is image-only. The alpha in GameAlpha-2.4K is recovered under controlled, matte-friendly condi-



(a) (b) (c)

Figure 7: **Transparency structure of GameAlpha-2.4K. (a)** Per-pixel alpha over all GameAlpha-2.4K videos $\times$ 97 frames at the training geometry (624×640; log scale): 70.1% of pixels are fully transparent ($\alpha \leq 2/255$; exactly 0: 69.2%) and 27.8% effectively opaque ($\alpha \geq 253/255$; exactly 1: 1.2%), leaving a 2.2% soft band ($2/255 < \alpha < 253/255$); the $\pm 2/255$ margins absorb video-codec noise. Under the naive definition $0 < \alpha < 1$ the soft fraction is 29.6%, so the band definition must accompany any quoted number. **(b)** Per-video fraction of fully transparent tokens under the conservative token criterion (16×16 spatial and 4-frame temporal max-pooling, token visible if any pixel in its receptive field is): mean 0.63, median 0.64. This is an oracle upper bound on the routable-token fraction; the validation-set mean initial routed fraction after evaluation 2 (0.35, dashed) lies below the fully transparent token fraction of 98.9% of videos. **(c)** Transparent-token sets of adjacent latent frames overlap almost perfectly (per-video mean IoU, median 0.95), whereas a static first-frame plan (visible set dilated by r=2) overlaps later-frame transparency far less (median 0.72): assets move and grow, and on average 12.2% of plan-frozen tokens become visible in a later frame. This gap underlies the ordering of the routing ablation in Table 2. Token statistics here use the $2/255$ threshold of the dataset audit; the router's transparency labels in Sec. 3.3 use $\eta = 0.05$ on decoded alpha — the two thresholds serve different objects and are not interchangeable.

Table 4: Comparison of reported training data for RGBA generation or video matting by scale, domain, and public availability. LayerDiffuse is included as an image-only reference. "Availability" reports release status: ✓/✗ denote publicly available/not publicly available at the time of writing, and "Planned" denotes a planned public release. n/d and n/a denote not disclosed and not applicable.

Dataset	Clips	Frames	Domain	Availability
VideoMatte240K	484	240.7K	human	✓
Wan-Alpha (DiT)	429	n/d	human/VFX	✓
TransVDM	∼10K	∼150K	objects	✗
TransAnimate	∼30K	n/d	game FX/objects	✗
TransText	16.1K	n/d	glyphs	✗
LayerDiffuse	1M imgs	n/a	general	✗
GameAlpha-2.4K	2,395	232,315	game assets	Planned

tions and retained only after staged gates (Appendix A.1), which is a different supervision contract from post-hoc matting of arbitrary composites. Finally, the planned release status and licensing for GameAlpha-2.4K, the accompanying code, and the model checkpoint are documented in Appendix F.

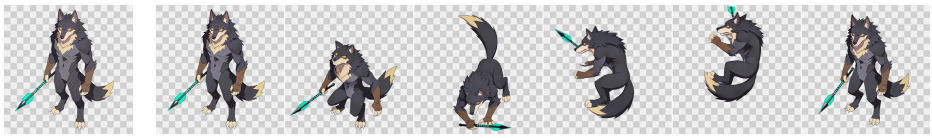
B Implementation and Protocol Details

B.1 DiT Adaptation and Inference Configuration

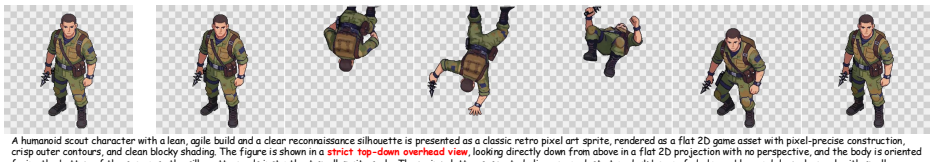
As stated in Sec. 4.1, we adapt Wan2.1-I2V-14B (Wan et al., 2025) with rank-32 DoRA (Liu et al., 2024b) modules on the attention projections (q, k, v, o) and feedforward layers, trained for 10 epochs with AdamW (Loshchilov et al., 2017) at 1.4×10^{-4} and weight decay 0.01 under distributed bfloat16. The per-GPU batch size is 1 without gradient accumulation (global batch 32). The shifted-



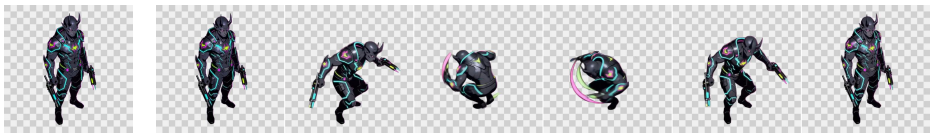
A humanoid guard character appears with a clear, readable protective silhouette, clad in polished steel armor over deep royal blue cloth layers, secured with dark brown leather straps and fittings. The design reads as disciplined and defensive, with grounded proportions, crisp plate shapes, and controlled hand-painted edge definition suited for a production-quality game sprite. The view is an isometric orthographic projection with a pseudo-3D game camera and a 2.5D angle, with no perspective convergence, and the character is oriented toward the screen bottom-right diagonal with the front-right side visible. The character holds a magical wand designed as a compact arcane focus, glowing violet along its core with clean metallic gold accents and a refined crafted finish. The wand feels functional and magically energized, visually distinct while remaining proportioned to read clearly in an isometric game asset. The character is performing a compact **combat dodge roll animation** while remaining centered in place, tucking the body into a controlled rotational roll and keeping the glowing violet wand secured in hand throughout the motion. The action feels defensive, efficient, and battle-ready, with readable armor and cloth overlap during the roll, brief but continuous violet glow visibility from the wand, and only minimal movement so the motion reads as a stable in-place loop. The overall style is a crisp 2.5D isometric game asset with a pre-rendered 3D sprite look, clean studio-like lighting, strong silhouette readability, clear material separation, polished hand-painted edge control, and subtle brush rhythm, so the final result reads cleanly as a high-quality 2D game sprite.



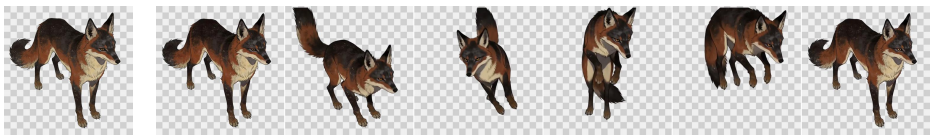
A humanoid werewolf is depicted as a modern anime-inspired 2D character illustration with cel-shaded rendering, flat controlled color blocking, crisp inked outlines, and consistent angular shape language. It has an athletic digitigrade build and is covered in charcoal gray and cool brown fur with creamy highlight accents along the muzzle, chest, forearms, and tail edge. The head is distinctly lupine with pointed ears, a strong snout, and glowing amber eyes, while the hands and feet remain expressive and humanlike in structure, tipped with pale claws. The silhouette is sleek and clearly readable, with sharp fur tufts and clean anatomical separation suited for stylized 2D character art. It is holding a streamlined dart with a sleek cel-shaded finish in neon teal and dark steel, featuring a narrow pointed tip, compact fletching, and crisp graphic paneling that reads clearly in a modern anime illustration. The character is shown in a strict **top-down overhead view**, looking directly down from above in a 2D flat projection with no perspective, and it remains oriented facing the bottom of the screen. It performs tight in-place **parkour flips and agile traversal motions in a continuous loop**, springing from compressed leg poses into compact aerial rotations, shifting through quick hand-assisted vault-like transitions, controlling the torso precisely through twists, and landing with sharp clawed precision that rebounds immediately into the next acrobatic move while staying centered in place, with the dart remaining clearly visible throughout.



A humanoid scout character with a lean, agile build and a clear reconnaissance silhouette is presented as a classic retro pixel art sprite, rendered as a flat 2D game asset with pixel-precise construction, crisp outer contours, and clean blocky shading. The figure is shown in a **strict top-down overhead view**, looking directly down from above in a flat 2D projection with no perspective, and the body is oriented facing the bottom of the screen so the silhouette reads instantly at small sprite scale. The main palette uses muted olive green, dusty tan, dark brown, faded navy blue, and deep charcoal, with small restrained dull silver-gray accents. The clothing and gear are practical and lightly worn, built from simple layered shapes that suggest a field-ready outfit, compact utility straps, sturdy boots, and a low-profile explorer look without excessive ornament. The character carries a small set of caltrops as compact metal hazard spikes, kept clearly visible as practical scout equipment rather than a large weapon; the caltrops read as simple dark metallic forms with tiny dull silver-gray highlights, attached to an easy-access pouch at the waist or held in a controlled, ready manner suited to a mobile scout. The action shows a compact in-place **parkour loop**, **cycling through tight acrobatic flips**, quick vault-like tucks, planted hand touches, and **agile rebound motions while staying centered on the starting position**. The body remains spring-loaded and controlled, with readable limb arcs, balanced landings, and brief crouched compression between beats to keep the motion physically grounded and naturally repeatable. The pixel treatment emphasizes strong silhouette separation, low-noise surfaces, simplified texture frequency, restrained cluster-based pixel placement, minimal anti-aliasing, and clear value grouping so the humanoid form, outfit layers, and carried caltrops remain legible throughout the motion.



A humanoid demon creature is shown with a strong, readable silhouette, facing toward the bottom of the screen with the front of the body visible, seen from a classic JRPG three-quarter top-down perspective in a high angle view that shows both the top and the front. It has dark metallic charcoal skin and fitted armor plates, combining infernal anatomy with sleek cybernetic design cues through sharp angular contours, layered plating, and luminous edge accents while remaining clearly demonic and humanoid. Electrified infernal markings run across the body and armor, glowing in neon cyan, magenta, and lime, integrated into a flat 2D game-art rendering style with crisp silhouette-first shape language, dark metallic massing, clean graphic surfaces, and controlled luminous accents for a cinematic but highly readable cyberpunk neon look. The creature is carrying sleek high-tech thieves' tools designed in black chrome, including compact lockpicks, probes, and precision implements accented with vivid neon cyan, magenta, and lime highlights. The tools look engineered, stealthy, and professional, matching the demon's dark metallic finish and glowing cyberpunk detailing. In motion, the humanoid demon performs a compact **combat dodge roll** in place, tucking the body into a fast defensive rotation and then recovering **smoothly back into a ready stance** without drifting from the original position. During the roll, the dark metallic armor and charcoal skin catch sharp neon cyan, magenta, and lime edge glows, while the electrified infernal markings briefly streak to emphasize motion in a smooth continuous looping action.



A beast-form fox is rendered as a compact, agile dark-fantasy creature with clearly vulpine anatomy, including a lean body, alert upright ears, a narrow muzzle, a thick tail, and grounded animal proportions suited for a 2D game asset. The main body is covered in dark russet fur, the limbs are soot-browned, the underfur is muted cream, and shadowy charcoal accents define the face, ears, tail, and lower silhouette, all treated in a gritty, desaturated dark-fantasy palette. **The view is a 3/4 top-down perspective** in a classic JRPG camera angle, seen from a high angle that shows both the top and front of the body, with the fox oriented diagonally toward the bottom-right of the screen and its front-right side more visible. The visual treatment is a grim, desaturated 2D illustration for a game asset, combining detailed ink linework with layered digital paint, restrained hand-painted edge control, visible brush rhythm, and strong material contrast between coarse outer fur, soft undercoat, and dusty surface grime. The shading remains physically grounded and readable for animation. The fox performs a compact in-place **jump**, crouching slightly before springing upward, with all four paws leaving the ground at once, the body tightening in midair and the tail counterbalancing the motion, then dropping straight back down to land in the exact same footprint with a soft rebound, completing a clean looping jump with no forward travel.

Figure 8: **GameAlpha-2.4K samples with their prompts.** Each row shows the RGBA reference (left), the video prompt (below; viewpoint and action constraints highlighted), and uniformly sampled frames of the clip (right); checkerboards denote transparency. The prompts are the structured output of the tuple sampler in Appendix A.1: the sampled entity, equipment, action, style, and viewpoint of Table 3 are each traceable in the text, and every action is constrained to be in-place and loop-closing.

Algorithm 1 Visibility-routed sampling (four evaluations). g: frozen transparency probe; margins $(r_s, r_t) = (2, 1)$; the first latent frame is never routed; with routing disabled (or on fallback) sampling is bit-exact to the dense sampler.

run evaluations 1–2 dense; keep $\hat{\mathbf{x}}_0^{[2]}$
$\ell \leftarrow g(\hat{\mathbf{x}}_0^{[2]})$; $P \leftarrow \mathrm{Dilate}_{r_s,r_t}(\{\ell_j \le \tau\}) \cup \{\text{frame } 0\}$; $F \leftarrow \complement P$
if $|F|/N < 10\%$ **or** $|\{j : |\ell_j - \tau| < \delta\}|/N > 15\%$ **return** dense sampling
store endpoints $\mathbf{x}^\star_{0,j} \leftarrow \hat{\mathbf{x}}^{[2]}_{0,j}$ and per-layer K/V of F {Cache-KV}
for $k = 3, 4$ **do**
    active tokens: fresh Q/K/V, attention over fresh + cached K/V, FFN
    routed tokens: advance by the x_0-lock transport (Eq. (7)) {no DiT pass}
    if $k = 3$: recompute P from fresh $\hat{\mathbf{x}}_0^{[3]}$ of active tokens; reactivate $F \cap P$ (fresh K/V on re-entry)
end for
return RGB and alpha decoded from the merged states

linear flow-matching scheduler uses shift 5, $\sigma_{\max} = 1$, and $\sigma_{\min} = 0$; training uniformly samples its 1,000 nonzero levels. Inference uses the four-evaluation LightX2V (Contributors, 2025) schedule described in Sec. 3.3, $\sigma = (1.000, 0.938, 0.834, 0.628)$ before the zero-noise endpoint, with FlowMatch shift 5, and generates 97 frames at 624×640.

B.2 Visibility Router Details

Probe. The transparency probe is a logistic regression fitted offline on dense denoising trajectories of training clips and frozen for routed inference. For each token, the feature flattens its 2×2 spatial patch across the 16 latent channels of $\hat{\mathbf{x}}_0^{[k]}$ into 64 dimensions; the label marks token j effectively transparent when the maximum decoded alpha over its spatiotemporal region Ω_j is at most $\eta = 0.05$ in the dense final output. The probe is a readout of the model's own evolving prediction, not an external saliency heuristic, and its cost is negligible relative to a DiT evaluation (a single $N \times 64$ matrix product per routing decision). Measured with the final checkpoint at inference precision (medians over repeated runs on one accelerator), reading the probe over all 39,000 tokens costs 0.8 ms per routing decision along the inference code path (0.05 ms once inputs are resident on the accelerator), decoding alpha from $\hat{\mathbf{x}}_0$ through the VAE alpha decoder costs 6.9 s, and one dense DiT evaluation costs 22.6 s: probe evaluation alone costs $\approx$0.004% of one evaluation, whereas an alpha-decoder read costs $\approx$31% of one evaluation. All routing operations are included in the end-to-end timings reported under the protocol in Appendix B.3.

Margins, gates, and reactivation. With $\tau = 0$, tokens with logit $\ell_j > \tau$ are predicted transparent; the complementary nontransparent support is dilated by $r_s = 2$ in space and $r_t = 1$ along latent time, and the first latent frame always stays active. Before any sparse state is constructed, a sample stays on the dense path if fewer than 10% of tokens are routing candidates or more than 15% are uncertain ($|\ell_j - \tau| < \delta$, $\delta = 2$); because this gate runs before any mutation, the fallback is bit-exact dense inference. After the first sparse evaluation, the protected support is recomputed from the fresh clean estimates of active tokens with the same margins, and any routed token reached by the updated support is reactivated for the final evaluation, re-entering at the current global noise level thanks to the x_0-lock transport.

B.3 Baseline Systems

All three RGB generators are fine-tuned on the GameAlpha-2.4K training split before evaluation. Wan uses the same DiT adaptation recipe and training budget as our model (Appendix B.1). Each generator receives the same white-composited reference, and all outputs follow the 97-frame evaluation protocol. MatAnyone2 (Yang et al., 2026) and SAM3 (Carion et al., 2025) are seeded with the same reference alpha used by our model; ours also encodes it in the RGBA latent anchor.

Generators. Wan2.1-I2V-14B is initialized from the official 480P weights and uses the same LightX2V CFG step-distillation LoRA as our backbone (4 evaluations, no negative branch, FlowMatch shift 5), generating natively at $624\times640 \times 97$. CogVideoX1.5-5B-I2V runs its official recipe (50 steps, dynamic guidance 6.0) at its native 768×768 and 16 fps — a one-to-one temporal mapping to the protocol — and is then resampled to 624×640. HunyuanVideo-I2V (13B) runs its official step count and guidance directly at the protocol resolution 624×640 (both sides divisible by 16); its 720p bucket is computationally infeasible at 97 frames, which we disclose rather than crop. All generator outputs are conformed to 624×640, 97 frames, 16 fps with high-quality H.264 (alpha read-back error $\approx 2 \times 10^{-4}$ MAE), and the Table 1 generators share seed 0.

Extractors. BiRefNet uses the RMBG-2.0 release (framewise, 1024^2 input, soft alpha); MatAnyone2 propagates from the first-frame dataset alpha (binarized seed mask); SAM3 tracks from the same seed and produces binary masks (largest connected component); UniVidX generates alpha with its released configuration.

Routing baselines. Planner constructs a fixed spatial routing mask from the reference alpha, whereas Union2D uses the temporal union of the dense model's final generated alpha as an offline static-support diagnostic. Both apply the same spatial mask across latent frames while retaining the first latent frame.

Runtime. Both tables report mean end-to-end wall-clock time per clip, excluding model loading. All variants in Table 2 use batch size 1 with the same checkpoint, input conditions, precision, resolution, frame count, and four-evaluation schedule. Timing covers condition encoding, denoising, RGB/alpha decoding, video encoding, and file writing, including probe evaluation, dilation, cache handling, token gather/scatter, and reactivation for routed variants. Table 1 additionally includes matting for two-stage systems, measured on identical hardware. Speedup in Table 2 is the dense mean runtime (125.9 s) divided by each variant's mean runtime. The paired three-seed timings are reported in Appendix C.5. These wall-clock ratios include the complete timed pipeline and are separate from token-evaluation saving fractions.

B.4 Metric Definitions

FVD. I3D features (Carreira & Zisserman, 2017) on checkerboard composites of RGBA outputs, following Unterthiner et al. (2019); Skorokhodov et al. (2021); both generated and reference sets use the same checker pattern and compositing code. The checker uses 16-pixel tiles with light/dark gray levels $230/179$ (XOR pattern, dark origin) at the evaluation resolution 624×640, identical for every frame and every system. For the GameAlpha-2.4K evaluation, FVD uses exactly 97 frames per clip and all $n{=}239$ validation clips (FVD values are not comparable across different n or frame counts); cross-domain cohorts are specified separately in Appendix C.7.

DINO-ID. Mean DINOv2 (Oquab et al., 2024) ViT-S/14 cosine similarity between 16 uniformly sampled generated frames (white composites) and the reference frame's white composite.

Aesthetic. Mean LAION aesthetic-predictor score over 8 uniformly sampled white-composite frames.

Motion smoothness. The VBench (Huang et al., 2024) motion-smoothness metric.

MAD. Mean absolute difference ($\times10^{-3}$) between generated and dataset alpha sequences (Lin et al., 2021a; Yang et al., 2026), over all pixels of all 97 frames at 624×640 (alpha decoded as the channel mean of the grayscale video). For GameAlpha-2.4K, the dataset alpha is extractor-selected supervision rather than renderer-native ground truth; consequently, MAD measures agreement with that reference sequence and does not by itself establish physical opacity accuracy or arbitrary-background recompositability.

Flow Difference. Following the adaptation described in Sec. 4.1: Farneback flows (Farnebäck, 2003) are computed on consecutive frames of the white-background RGB composite and of the alpha sequence; the endpoint distance between the two flows is averaged inside a dilated foreground

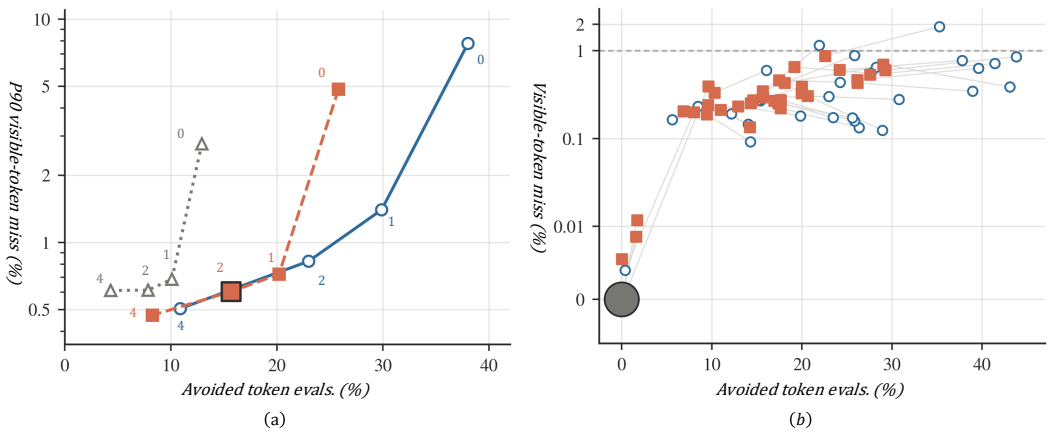


Figure 9: **Visibility-routing risk–yield trade-off.** The scan uses GameAlpha training trajectories. (a) The 90th-percentile final-visible-token miss rate versus the median fraction of token evaluations skipped across all four evaluations. Blue circles, orange squares, and gray triangles indicate routing after evaluations 1, 2, and 3, respectively; labels indicate r_s, and the black outline identifies the selected setting, $(k_{\text{route}}, r_s) = (2, 2)$. (b) Per-clip changes when routing is delayed from evaluation 1 to 2 at $r_s = 2$; lines connect the same clip, the gray marker summarizes clips with no savings, and the dashed line indicates a 1% miss rate.

mask and normalized by the sum of the mean magnitudes of both fields. Flow Difference diagnoses RGB–alpha motion alignment and is interpreted jointly with FVD (a frozen-content clip can have excellent alignment). Precisely, for each consecutive pair $(t{-}1, t)$: Farneback flows (parameters $0.5, 3, 15, 3, 5, 1.2, 0$) are computed on the grayscale white composite and on the alpha map; the foreground mask is the union $\max(\alpha_{t-1}, \alpha_t) > 0.05$ dilated twice with a $5{\times}5$ kernel (≈ 4 pixels); the pair score is $\text{mean}_{\text{fg}}\|F_{\text{rgb}} - F_\alpha\|_2 \,/\, (\text{mean}_{\text{fg}}\|F_{\text{rgb}}\| + \text{mean}_{\text{fg}}\|F_\alpha\| + 10^{-3})$; clip scores average the 96 pairs (empty-foreground pairs are skipped), and table entries average all 239 clips.

C Additional Experiments and Analyses

C.1 Routing-Time Risk–Yield Trade-off

Figure 9 reports the training-split risk–yield analysis underlying the configuration choice in Section 4.5. For N tokens per evaluation, each per-clip saving fraction is the skipped token-evaluation count divided by $4N$. This denominator includes the dense warm-up evaluations; the figure reports the median across clips. At $r_s = 2$, delaying routing from evaluation 1 to 2 reduces P90 final-visible miss from 0.824% to 0.603%, while median full-trajectory saving decreases from 22.99% to 15.68%. These token-update statistics are distinct from the measured end-to-end runtime and speedup in Table 2.

C.2 Early Readability of Token Transparency and Routing Time

The visibility criterion in Eq. (6) is analytic at the final output, but membership in the final-transparent set is unknown while denoising is still running. This section tests whether the model's step-wise clean estimates already contain a cheaply readable signal of *final* token transparency, and when that signal becomes reliable enough to route computation. From held-out denoising trajectories that pass an alpha-health gate (rejecting degenerate alpha predictions), we score every token at each of the four evaluations with the probe fitted on training trajectories and evaluate the same scores against two label sets: **Final-Transparency Prediction AUROC** against the final decoded alpha (does the current state predict the final transparent set?), and **Same-Step Readout AUROC** against alpha decoded from the same-step $\hat{\mathbf{x}}_0^{[k]}$ (does the probe correctly read the model's current transparency estimate?). Alpha maps are max-pooled to the DiT token grid; a token is visible if any alpha in its receptive field exceeds $\eta = 0.05$.

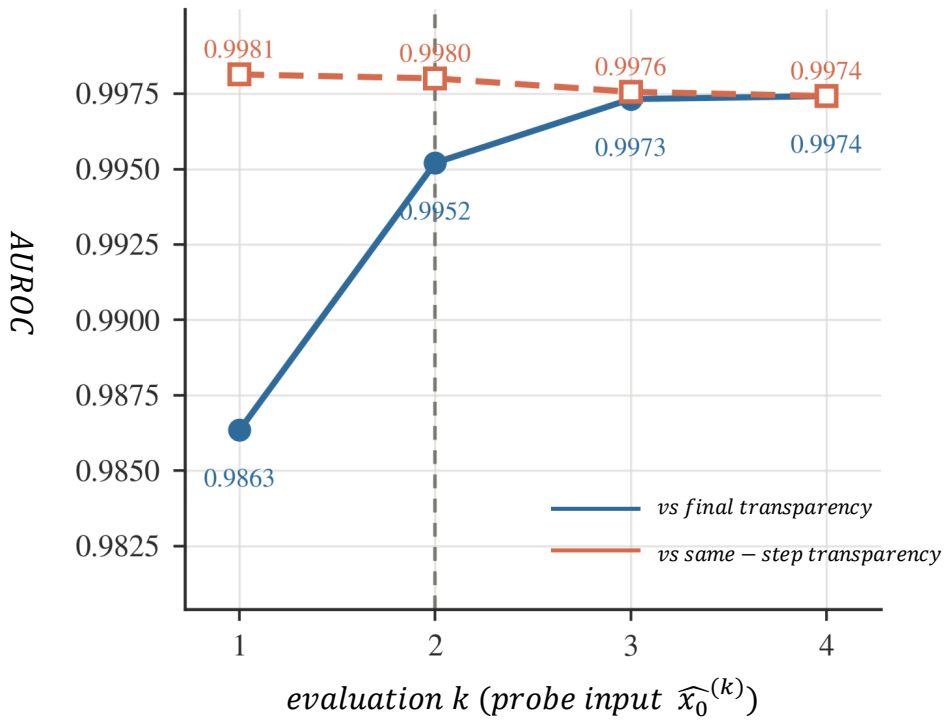


Figure 10: **Early readability of token transparency** (probe fitted on training trajectories; 48 held-out validation trajectories; 1,872,000 tokens per evaluation; identical token set for both curves; the alpha-health gate fired on 0 of 48 clips). Blue (solid, circles): probe scores against *final* transparency labels. Coral (dashed, squares): the same scores against *same-step* decoded alpha. The vertical line marks the routing point k=2. Same-step readout stays at 0.9974–0.9981 from the first evaluation, while final-transparency prediction rises from 0.9863 (evaluation 1) to 0.9952 (evaluation 2) and plateaus near 0.997; at evaluation 4 the two targets coincide by construction. The early gap is consistent with belief drift of the model's own alpha estimate rather than a readout failure.

Two observations follow. First, the probe reads the model's current transparency estimate essentially from the first evaluation (readout 0.9981 at evaluation 1), so early routing is not limited by readout capacity. Second, the current estimate itself keeps converging toward the final output: final-transparency AUROC improves markedly from evaluation 1 to evaluation 2 (0.9863→0.9952, closing the gap to the readout curve from 0.0118 to 0.0028) and only marginally afterwards (0.9973 at evaluation 3), which supports the training-selected routing point $k_{\text{route}} = 2$ — waiting until evaluation 3 buys a small ranking gain but halves the sparse suffix. The held-out AUROC analysis supports early readability of final transparency; the deployed routing time and margin are selected using the training-split risk–yield analysis in Appendix C.1. The spatiotemporal margin, local reactivation, and pre-routing dense fallback in Sec. 3.3 mitigate routing errors. AUROC measures ranking quality rather than classification accuracy.

C.3 How the Alpha Trajectory Evolves

This analysis visualizes *why* early routing plus a local margin is the right shape of solution: the model's early clean estimates already approximate the final generated silhouette, and the remaining changes concentrate near the foreground boundary as contours, soft edges, and motion details emerge.

Reading Figure 11 column by column: at evaluation 1 the silhouette already approximates that of the final generated output, with residual coral/blue disagreement forming a band around the boundary (row 3 shows the largest early drift among the three examples); by evaluation 2 the band thins to one or two token widths; evaluations 3–4 refine soft edges.



Figure 11: **Evolution of decoded alpha estimates** (deployed checkpoint; dense validation trajectories). Three manually selected alpha-healthy clips (rows) at evaluations 1–4 (numbered columns) and the final output (rightmost column), decoded directly from the stored clean-estimate latents. Grayscale shows the current decoded alpha; coral marks support that *appears later* (missing relative to the final alpha), and blue marks support *present now but absent at final*. Rows are cropped independently, so scales are not comparable across rows. The visualization diagnoses trajectory self-consistency — how the model's own estimate converges to its own final output — and does not measure matte accuracy against an external alpha target.

C.4 Within-Frame Support-Mismatch Distances

Figure 12 asks where the evaluation-2 transparency estimate disagrees with the final output within each frame: most raw mismatches lie within one or two token cells of the current foreground boundary, supporting the training-selected local margin ($r_s = 2$) over a globally enlarged active set. The longer-distance mismatches motivate complementary safeguards: uncertainty-based dense fallback and local revision of the routed set as the estimated foreground evolves (Appendix C.5).

C.5 Multi-Seed Evaluation and Matched Reactivation Ablation

We evaluate dense inference, Cache-KV routing, and a matched no-reactivation variant on all 239 GameAlpha validation inputs using seeds 0, 1, and 2, yielding 2,151 generated videos. All variants share the checkpoint, reference and text conditions, and sampling schedule, with identical initial noise for each input–seed pair. The no-reactivation variant uses the same visibility scores, evaluation-2 routed set, spatial–temporal safety margins, dense-fallback decisions, x_0-lock, and Cache-KV implementation as our router, differing only in whether routed tokens can return to the active set.

Table 5 reports FVD under the checkerboard protocol in Appendix B.4 and mean end-to-end time across all 717 outputs per variant. Timing follows Appendix B.3.

Across three seeds, Cache-KV achieves FVD 174.15 ± 0.77, compared with 172.75 ± 1.73 for dense inference, while reducing mean end-to-end latency from 125.10 to 104.34 seconds per video ($1.20\times$ speedup). Disabling reactivation yields FVD 177.22 ± 0.90 at 104.05 seconds. In the paired three-seed evaluation, reactivation reduces mean FVD from 177.22 to 174.15, an improvement of 3.07, with 0.29 additional seconds per video. Here, $\pm$ denotes the standard deviation across seeds. Appendix C.6 reports deployment frequencies, and Appendix D.2 presents paired examples from this matched evaluation.

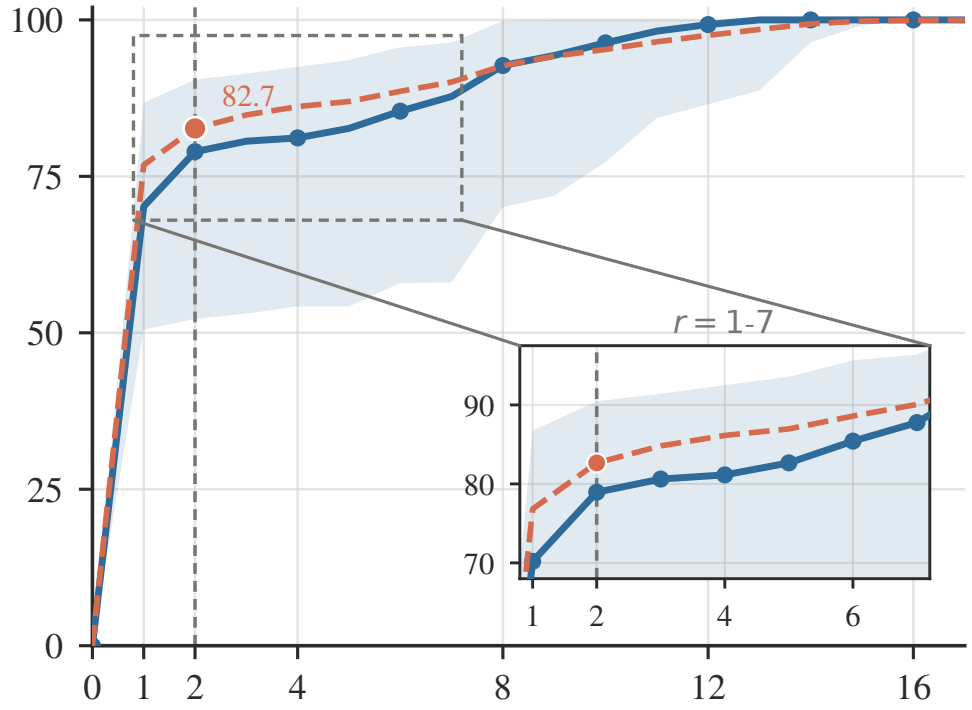


Figure 12: **Within-frame support-mismatch distance CDF (pilot).** For alpha-healthy analysis trajectories at k=2: cumulative fraction of raw per-frame support mismatches (tokens transparent in the evaluation-2 estimate but visible in the final output) within a given 2D Chebyshev distance of the current foreground, *before* any dilation or first-frame protection is applied. Blue: per-sample median CDF; band: P10–P90 across samples; coral dashed: pooled over tokens (82.7% of pooled mismatches lie within radius 2, marked). The inset enlarges r = 1–7, where the median and pooled curves differ most. Per-sample mismatch counts vary widely, so the pooled curve is weighted toward high-mismatch samples. This is a mechanism diagnostic of the training-selected margin, not a measurement of the deployed 3D margin: it cannot be read as "r_s=2 covers 82.7% of routed misses".

Table 5: Multi-seed evaluation on GameAlpha-2.4K (n = 239 per seed). FVD mean and standard deviation are computed across three paired seeds; time is the mean complete inference time per video, including video encoding and file writing. Cache-KV and no reactivation differ only in local reactivation.

Variant	FVD mean ± SD↓	End-to-end time (s)↓
Dense	172.75 ± 1.73	125.10
Cache-KV	174.15 ± 0.77	104.34
No reactivation	177.22 ± 0.90	104.05

C.6 Fallback and Reactivation Statistics

Figure 13 measures how often the router's safety mechanisms actually fire on the validation split. *Dense fallback.* The pre-routing gate keeps 7 of 239 clips (2.9%) on the dense path: 4 trigger the low-support condition (fewer than 10% candidates), 2 the high-uncertainty condition (more than 15% uncertain tokens), and 1 both. Because the gate decides before any sparse state is constructed, these clips execute bit-exact dense inference. *Routed fraction and full-trajectory saving.* The 35% is the validation-set mean initial routed fraction after evaluation 2. The 15.68% is the median four-evaluation token saving from the training-split scan in Appendix C.1, including the two dense evaluations in its denominator. These statistics differ in data split, denominator, and aggregation; neither is a wall-clock reduction. *Reactivation.* Every routed clip reactivates tokens between evaluations 3–4 (median 325 tokens or 2.1% of the routed set; maximum 16.4%). The mean of 360 reactivated

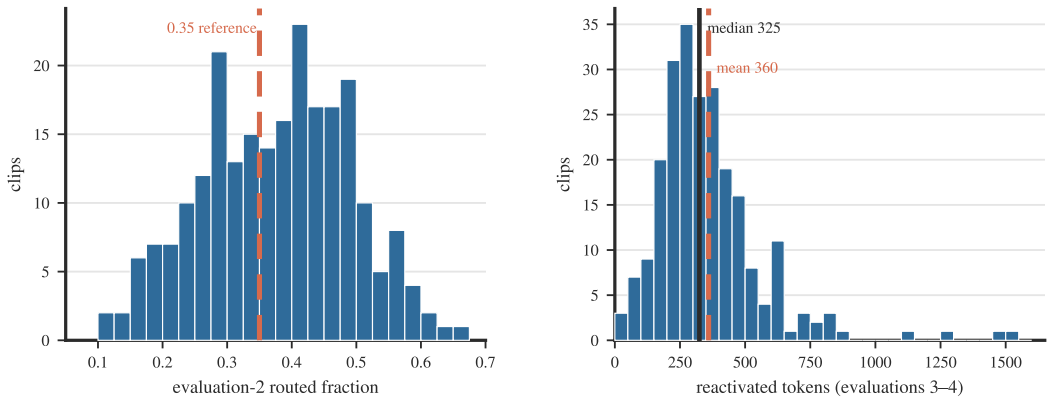


Figure 13: **Deployment statistics of the safety mechanisms** (validation split, n=239, primary realization). Left: per-clip distribution of the evaluation-2 routed fraction over the 232 routed clips (dashed: the validation-set mean initial routed fraction of 0.35 reported in Table 2). Right: per-clip reactivated-token counts between evaluations 3–4 (solid: median 325; dashed: mean 360).

tokens corresponds to 0.46 percentage points of the sparse suffix ($2 \times 39{,}000$ token evaluations per routed clip).

C.7 Cross-Domain Evaluation on VideoMatte240K

This section evaluates cross-domain behavior on VideoMatte240K (Lin et al., 2021a): human-subject clips outside the game-asset training domain of our Stage-2 generator. For this cross-domain diagnostic, we use a separately trained RGB-A VAE checkpoint that was trained without VideoMatte240K; none of the evaluated VideoMatte240K clips or alpha sequences was used to train that VAE, the Stage-2 DiT, or the transparency probe. For each clip, we take the first 97 source frames without temporal resampling and apply to foreground and alpha a 421×432 crop centered on the union of ground-truth alpha support ($\alpha > 2/255$) over those frames. We then resize both to 624×640 and encode them at 16 fps. All systems receive the resulting white-composited first frame and the same per-clip caption from a fixed open captioning model; MatAnyone2 and SAM3 also receive the same first-frame ground-truth alpha. Our model is evaluated with dense inference and with visibility routing at $(k_{\text{route}}, r_s) = (2, 2)$, selected on the GameAlpha training split and applied unchanged. Neither routing parameters nor checkpoints are selected using the VideoMatte240K results. The same configuration also performs best among the routing settings evaluated on VideoMatte240K, providing further evidence that the configuration selected on the GameAlpha training split transfers across domains. Metrics follow Sec. 4.1, with MAD computed against the dataset's ground truth alpha.

Table 6 reports the grid. Among the directly comparable full-cohort rows, our model obtains the lowest FVD (125.7 routed; 129.7 dense) and MAD (105.2 routed; 105.0 dense). With the shared Wan backbone, generate-then-matte variants reach FVD 132.6–135.9 and MAD 127.9–135.9. These are full-system comparisons: RGB-A adaptation and routed inference vary together, so the table does not isolate a causal gain from joint foreground-and-alpha generation alone. Flow Difference must still be read with content quality: Wan–SAM3 has a slightly lower value than our router (0.618 vs. 0.621) but a higher FVD (135.9 vs. 125.7).

Under Stage-2 domain shift, dense fallback rises from 2.9% in-domain to 5.2% (25/484: 22 low-support and 3 high-uncertainty clips). Among clips that enter sparse execution, every clip reactivates tokens (median 198 tokens, versus 325 in-domain). Routed and dense results remain close across the reported metrics, demonstrating that the deployed routing configuration transfers to these external inputs while retaining comparable generation quality.

Table 6: Cross-domain comparison on VideoMatte240K. Full-cohort rows use n=484 and 97 frames; MAD uses ground-truth alpha. The separately trained RGB-A VAE checkpoint used in this diagnostic was trained without VideoMatte240K. †UniVidX rows use the same seeded 400-clip manifest prefix due to per-clip cost, so their FVD is not comparable with the n=484 rows. DINO-ID/Aesthetic use 447–483 and 395–400 valid scores for the n=484 and n=400 rows, respectively, after auxiliary NIQE failures; Flow-Diff uses 480–484 clips for full-cohort rows (400 for UniVidX) where foreground-flow support is available. **Bold** marks the best n=484 value per column, excluding the ground-truth anchor.

Generator	Matte	FVD↓	DINO-ID↑	Aes.↑	Motion Sm.↑	Flow-Diff↓	MAD (10^{-3})↓
Wan2.1-I2V (4-step)	BiRefNet	133.9	0.887	5.166	0.991	0.624	135.9
	MatAnyone2	132.6	0.888	**5.168**	0.991	0.619	131.7
	SAM3	135.9	0.885	5.149	0.991	**0.618**	127.9
	UniVidX†	143.8	0.866	5.155	0.992	0.633	138.5
CogVideoX1.5-I2V	BiRefNet	179.5	0.860	4.868	0.988	0.640	128.3
	MatAnyone2	175.2	0.865	4.876	0.988	0.621	125.3
	SAM3	176.4	0.859	4.872	0.987	0.691	128.2
	UniVidX†	178.2	0.854	4.870	0.988	0.652	128.3
HunyuanVideo-I2V	BiRefNet	423.3	0.659	4.793	0.992	0.701	283.4
	MatAnyone2	372.2	0.697	4.730	0.993	0.670	234.6
	SAM3	353.6	0.719	4.739	0.991	0.668	245.0
	UniVidX†	341.6	0.581	4.582	0.993	0.617	225.4
Ours (dense)		129.7	0.905	5.119	**0.994**	0.624	**105.0**
Ours (visibility router)		**125.7**	**0.906**	5.107	**0.994**	0.621	105.2
Ground truth (anchor)		≈0	0.885	4.930	0.990	0.588	—

Table 7: **Controlled loss-weighting ablation** on the GameAlpha-2.4K validation set (n=239). The uniform row is the dense reference reported in Table 2. The alpha-weighted model has the same 36-channel architecture and training recipe; only Eq. (8) changes. Both models are evaluated with dense four-step inference. **Bold** marks the better value; ties at displayed precision are both bolded.

Training objective	λ	FVD↓	DINO-ID↑	Aesthetic↑	Motion Sm.↑	Flow-Diff↓	MAD (10^{-3})↓
Uniform	1.0	**172.3**	**0.840**	**5.317**	**0.988**	**0.550**	**108.2**
Alpha-weighted	0.2	185.8	0.836	5.307	**0.988**	0.560	108.5

C.8 Loss-Weighting Ablation

To test whether visibility should also reweight the training objective, we retrain a controlled variant of the same 36-channel architecture while changing only the spatial weighting of the flow-matching loss. For latent cell i, let $a_i \in [0,1]$ denote the aligned dataset alpha occupancy and define

$$w_i = \lambda + (1-\lambda)a_i, \quad \mathcal{L}_\lambda = \mathbb{E}\left[\omega(t)\frac{\sum_i w_i \|\hat{\mathbf{v}}_{t,i} - \mathbf{v}_i\|_2^2}{16\sum_i w_i}\right]. \tag{8}$$

The weight is shared by all 16 channels of the merged RGB–alpha latent. Thus, $\lambda = 1$ recovers the spatially uniform objective, whereas $\lambda = 0.2$ assigns a fully transparent cell one fifth of the weight of an opaque cell and interpolates continuously through partial opacity. Normalization by the mean weight keeps the overall loss scale comparable. All data and optimization settings are unchanged. Table 7 evaluates both models with dense inference so that routing does not confound the comparison.

Table 7 shows that downweighting transparent regions raises FVD from 172.3 to 185.8 and reduces DINO-ID from 0.840 to 0.836, while motion smoothness is unchanged at the reported precision. Flow Difference increases from 0.550 to 0.560, while MAD against the dataset alpha increases from 108.2 to 108.5. Under this controlled setting, reallocating loss toward visible cells therefore degrades distribution-level video quality without improving RGB–alpha alignment or agreement

Table 8: Blinded human evaluation on 20 cases by 10 professional data annotators. Each method receives 200 ratings per criterion. Percentages sum to 100 within each method and criterion. The evaluated cases form an enriched qualitative pool rather than a random sample of the full validation set.

	Visual Attractiveness (%)			RGBA Alignment (%)		
Method	**Low**	**Mid**	**High**	**Low**	**Mid**	**High**
Wan2.1-I2V + BiRefNet	12.0	30.0	58.0	17.5	28.0	54.5
CogVideoX1.5-I2V + BiRefNet	81.5	16.0	2.5	7.5	40.0	52.5
HunyuanVideo-I2V + BiRefNet	67.0	28.5	4.5	31.0	28.0	41.0
Ours	11.0	27.0	**62.0**	4.5	19.5	**76.0**

with the extractor-selected alpha target, supporting the spatially uniform objective used in the main model. This result is specific to the tested $\lambda = 0.2$ weighting and does not rule out every alternative weighting schedule.

C.9 Human Evaluation

Participants. Ten professional data annotators participated in a blinded, within-subject evaluation. Each participant independently rated all evaluated outputs, and participant identities are anonymized as P01–P10 in the analysis.

Protocol. We evaluate 20 reference–prompt pairs. For each pair, participants view the checkerboard composites produced by four systems: Wan2.1-I2V-14B (4-step) + BiRefNet, CogVideoX1.5-5B-I2V + BiRefNet, HunyuanVideo-I2V + BiRefNet, and our deployed visibility-routed RGBA model. All methods are hidden behind labels A–D. Label positions are counterbalanced over the 20 cases, such that every method appears five times at each position. Each participant therefore rates 80 outputs on two criteria using a three-level ordinal scale (low, medium, or high).

The two criteria displayed on the rating sheet were (1) *Visual Attractiveness*, which asks for the visual appeal of each generated checkerboard composite, and (2) *RGBA Alignment*, whether RGB and alpha align correctly. The latter wording follows the RGBA Alignment dimension used by TransPixeler (Wang et al., 2025); our first criterion is not TransPixeler's Motion Quality dimension, which instead evaluates whether generated motion matches the corresponding text description. We therefore report the rating-sheet construct under its literal name and do not claim to replicate TransPixeler's evaluation dimensions or pairwise-preference protocol. Checkerboard ratings provide a perceptual measure of visible RGBA alignment, not pixel-accurate alpha quality or arbitrary-background recompositability.

Analysis. We map low, medium, and high to 1, 2, and 3 only for descriptive summaries. The participant is the independent unit: for each method and criterion, we first average over the same 20 cases within each participant. We then compare our method against each baseline using exact two-sided Wilcoxon signed-rank tests on the 10 paired participant summaries and apply Holm correction over the three baseline comparisons within each criterion. We additionally report the complete ordinal-response distributions in Table 8. Ordinal Krippendorff's alpha is .679 for Visual Attractiveness and .508 for RGBA Alignment.

Results. For Visual Attractiveness, our method obtains the largest share of high ratings (62.0%), followed by Wan2.1 + BiRefNet (58.0%). Participant-level comparisons show that our method receives higher ratings than CogVideoX1.5 + BiRefNet and HunyuanVideo + BiRefNet (Holm-adjusted $p = .0059$ for both), while no significant difference is observed relative to Wan2.1 + BiRefNet ($p = .5508$).

For RGBA Alignment, our method achieves the largest share of high ratings (76.0%), substantially exceeding Wan2.1 + BiRefNet (54.5%), CogVideoX1.5 + BiRefNet (52.5%), and HunyuanVideo + BiRefNet (41.0%). Participant-level comparisons further show that our method receives significantly higher RGBA Alignment ratings than all three generate-then-matte baselines: Wan2.1 +

Table 9: Paired reference-alpha source ablation on all 239 GameAlpha validation inputs (seed 0). Both conditions use the same dense RGBA generator. Time is mean end-to-end seconds per video and includes alpha estimation for RGB-and-text-only inputs.

External inputs	**Reference alpha**	**FVD↓**	**Time (s)↓**
RGB + text + reference alpha	Dataset-provided	171.56	125.34
RGB + text	Estimated from RGB	172.79	126.55

BiRefNet (Holm-adjusted $p = .0254$), CogVideoX1.5 + BiRefNet ($p = .0117$), and HunyuanVideo + BiRefNet ($p = .0059$).

Overall, our method achieves the strongest perceptual evaluation profile: it obtains the highest high-rating shares for both criteria, significantly improves RGBA Alignment over all three generate-then-matte baselines, and receives significantly higher Visual Attractiveness ratings than CogVideoX1.5 and HunyuanVideo, while its paired comparison with Wan2.1 is not significant.

C.10 Paired Ablation of Reference Alpha Source

Protocol. We vary the source of the first-frame alpha while holding the RGBA generator fixed. All 239 GameAlpha validation inputs are evaluated under both conditions with seed 0, yielding 239 pairs and 478 generated videos. Both conditions use the same checkpoint, RGB-A VAE, four-evaluation LightX2V schedule, RGB reference, text, and initial noise. Each pair runs on the same physical GPU at batch size 1 in bfloat16, generating 97 frames at 624×640 with routing disabled.

The provided-alpha condition uses the dataset's first-frame reference alpha. The RGB-and-text-only condition estimates alpha solely from the same white-composited RGB reference using the frozen RMBG-2.0 release of BiRefNet (Zheng et al., 2024); no dataset alpha is supplied to this condition during inference. Alpha estimation is performed only when constructing the first-frame condition; the video generator subsequently produces RGB and alpha jointly.

Both conditions use the same compositing background and FVD implementation. Mean end-to-end time follows Appendix B.3, including reference preparation, alpha estimation when used, condition encoding, denoising, RGB/alpha decoding, video encoding, and file writing; model loading is excluded.

Results. Table 9 reports FVD 172.79 for RGB-and-text-only inputs, an increase of 1.23 relative to supplied reference alpha, with 1.21 additional seconds of mean end-to-end runtime. This paired evaluation demonstrates RGBA video generation from RGB-and-text-only external inputs using the same fixed generator.

D Additional Qualitative Results

D.1 Full Generator–Extractor Grid

Together with the qualitative comparison in Figure 2, Figures 14 and 15 cover the three RGB generators and four post-hoc extractors in the quantitative grid. Within each example, the paired rows use the same reference and prompt.

D.2 Qualitative Routing Ablation

Figures 16 and 17 complement the quantitative routing ablation in Sec. 4.4 with selected paired examples.

D.3 Prompts of the Qualitative Examples

The qualitative comparison in Figure 2 pairs three RGB generators with BiRefNet or MatAnyone2. Within each group, our model and the corresponding two-stage pipeline receive the identical



Figure 14: **Additional qualitative comparison with SAM3.** Each example shows the prompt (left), common RGBA reference (center), and four frames at the same temporal indices from the indicated RGB generator followed by SAM3 (top) and from our single-stage model (bottom). Row pairs use Wan2.1-I2V, CogVideoX1.5-I2V, and HunyuanVideo-I2V from top to bottom; checkerboards denote transparency.

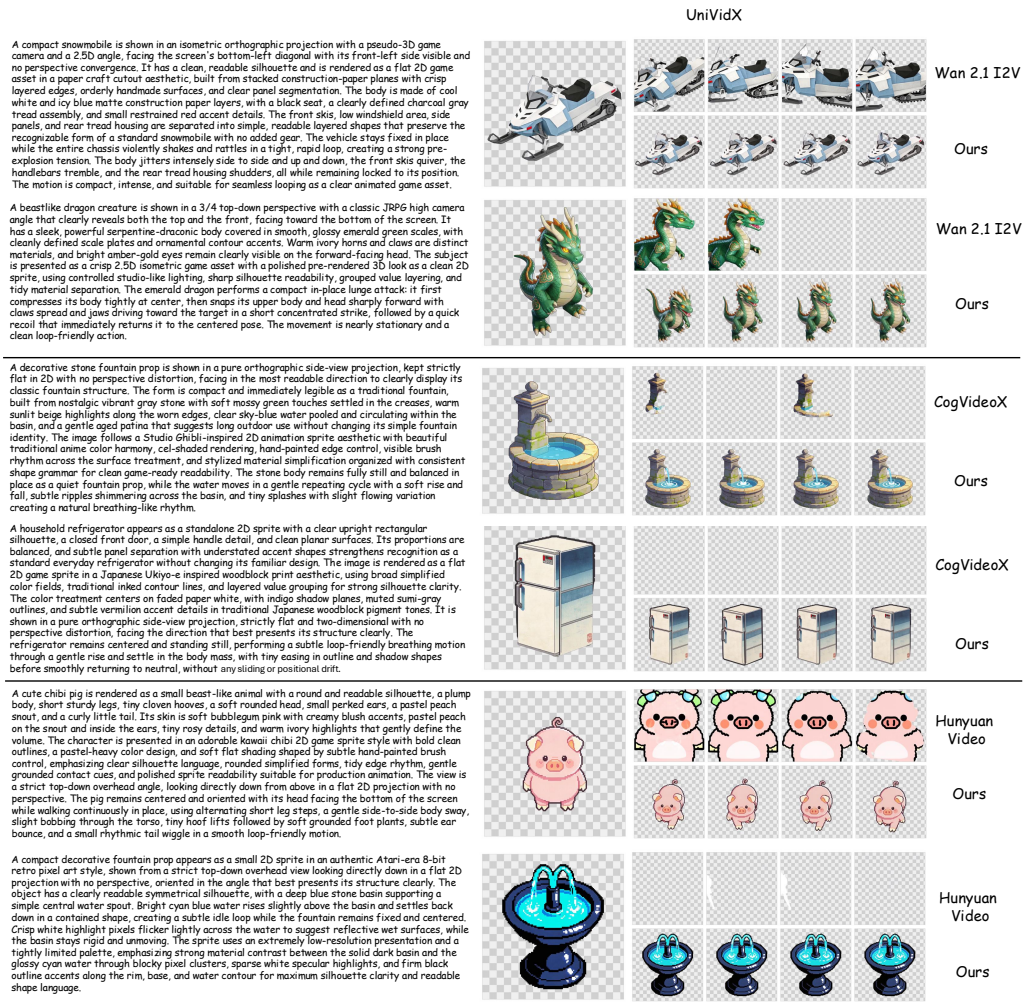


Figure 15: **Additional qualitative comparison with UniVidX.** The layout follows Figure 14: the upper and lower rows show the same four temporal indices from the indicated RGB generator followed by UniVidX and from our model, respectively, under the same prompt and reference. Checkerboards denote transparency; checkerboard-only panels contain no visible foreground in the displayed RGBA output.

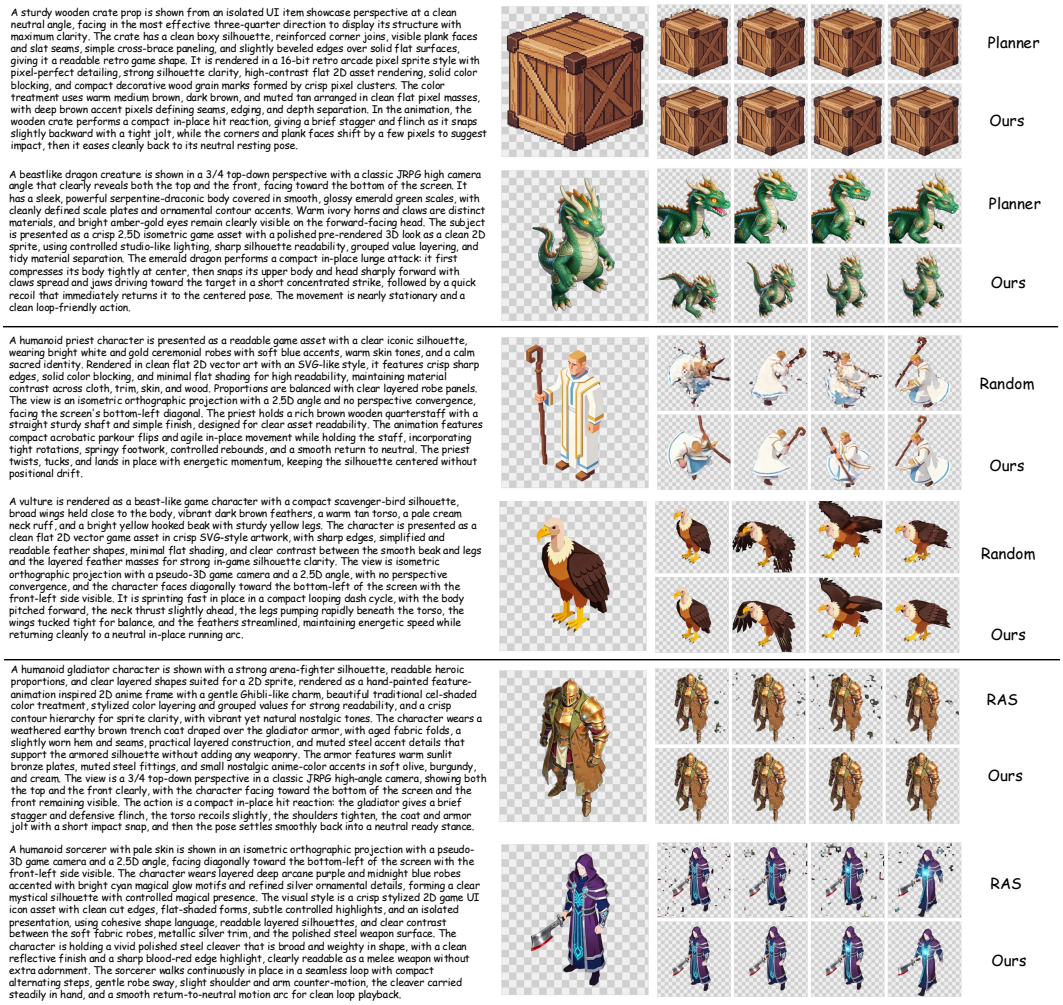


Figure 16: **Qualitative routing ablation: selection policies.** Each example shows the prompt (left), common RGBA reference (center), and the same four temporal indices from the named alternative (top) and our full visibility router (bottom). Two cases each compare Planner, Random, and RAS under the matched realized-skip protocol reported in Sec. 4.4. Checkerboards denote transparency.

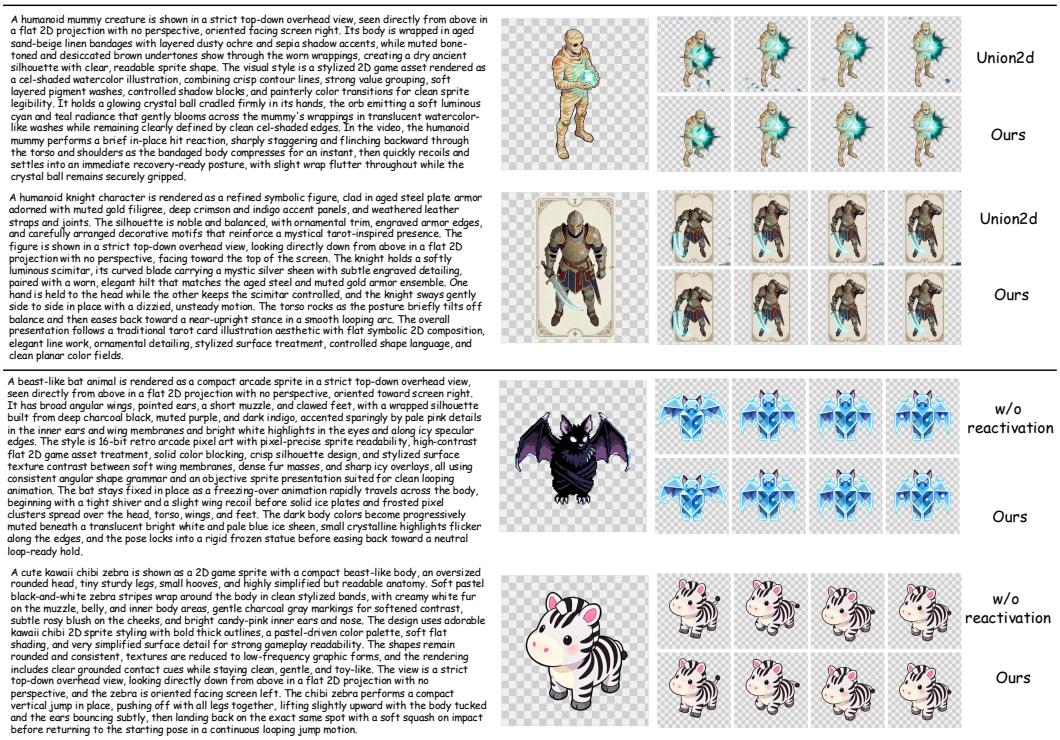


Figure 17: **Qualitative routing ablation: static support and reactivation.** The layout follows Figure 16. Union2D follows the same matched realized-skip protocol, whereas the no-reactivation variant shares our evaluation-2 routing decision but fixes the routed set through evaluations 3–4. The bat and zebra cases use seed 0 from the matched evaluation in Appendix C.5, with zero-based frame indices $24, 48, 72, 96$; the top and bottom rows show no reactivation and Cache-KV with reactivation, respectively.

text prompt reproduced below and the identical white-composited reference frame; the prompts are quoted verbatim, and the sampled entity, equipment, action, style, and viewpoint of Table 3 remain traceable in each of them.

Top row, left — Wan2.1-I2V (4-step) + BiRefNet. A rugged humanoid barbarian with weathered tan skin, a sturdy survivalist silhouette, and a battle-worn presence — grounded proportions and practical anatomy, a believable fantasy warrior rather than an exaggerated monster. Rendered as hand-drawn 2D game concept art with clean readable linework, controlled contour hierarchy, and painterly digital rendering, using expressive brushwork and stylized textures to define materials clearly while preserving strong sprite legibility and cohesive shape language. Strict top-down overhead view looking directly downward in flat 2D projection with no perspective, the character facing the bottom of the screen. He wears a worn mask of carved bone or aged wood, hand-shaped with natural brush-painted markings and a scuffed surface. The outfit is rugged earth-toned leather with muted iron and bronze accents — functional, distressed, repeatedly repaired, no weapon. He continuously climbs a vertical rope in place in a seamless loop, alternating hand-over-hand pulls with compact leg engagement and controlled body lift while staying centered. The motion is grounded and rhythmic, with subtle torso compression and extension, slight movement in the leathers and straps, and the mask stable as he ascends without changing screen position.

Top row, right — Wan2.1-I2V (4-step) + MatAnyone2. A humanoid vampire creature is shown in a strict top-down overhead view, seen directly from above in a flat 2D projection with no perspective, oriented toward screen left. It has pale, ashen skin, sharp predatory facial features, and deep crimson accents integrated into its design. The body is lean and undead-looking, with a clear readable silhouette and gothic menace. Layered shadowy black and burgundy garments frame the form, emphasizing clawed hands, tense posture, and a cold nocturnal presence. Over this clothing it wears a worn dark bulletproof tactical vest fitted close to the body, scuffed, utilitarian, and slightly weathered, rendered in natural hand-painted tones with stylized digital brush texture while remaining clearly protective modern armor rather than decorative costume. The image is rendered as hand-drawn 2D game concept art with clean controlled linework, painterly digital rendering, and expressive brush-driven surface treatment, using strong silhouette readability, cohesive shape language, and stylized but natural material definition suited for a polished 2D game asset, with crisp forms, selective texture, and a cinematic yet objective presentation. The humanoid vampire performs a compact vertical jump in place, springing upward and landing back in the exact same spot with a clean return to a neutral pose. During the lift and descent, the garments and the worn dark tactical vest react subtly to the motion while the body remains centered, the silhouette stays stable and readable, and the landing resolves into a clear idle-ready stance.

Middle row, left — CogVideoX1.5-I2V + BiRefNet. A cute chibi oni demon creature is shown in a strict top-down overhead view, looking directly down from above in a flat 2D projection with no perspective, facing the top of the screen. It has a compact humanoid body, an oversized head, a tiny torso and limbs, and a clean readable silhouette. Its skin is pastel cherry red, with small baby pink horns, creamy ivory fangs peeking from a playful mouth, round bubblegum pink cheeks, and soft lavender shadow accents defining simple stylized forms. The visual treatment is a kawaii chibi 2D sprite style with thick crisp outlines, a pastel candy palette, soft flat shading, minimal but polished surface detail, and strong shape-language consistency for clear game-asset silhouette readability. It is holding a candy-colored crossbow sized for the chibi character, with a chunky readable shape, a mint green body, baby blue limbs and grip details, and pale yellow accent pieces, built with simplified 2D game-friendly construction, soft flat color blocks, and bold outline clarity. The action is an in-place shooting animation: the chibi oni plants its stance, raises the mint-and-blue crossbow to aim, gives a compact firing snap with a small recoil through the arms and shoulders, and then smoothly returns along a clean arc to a neutral ready pose.

Middle row, right — CogVideoX1.5-I2V + MatAnyone2. A gargoyle creature with a beast-like silhouette is shown in an isometric orthographic projection with a pseudo-3D game camera at a 2.5D angle, with no perspective convergence, facing diagonally toward the bottom-right of the screen with its front-right side clearly visible. It has carved-stone anatomy and rugged architectural forms, broad clawed limbs, heavy wings, a horned head, and a snarling face. Its surface is weathered stone gray with cool blue undertones, broken by mossy olive staining, soot-dark recessed cracks, chipped edges, and muted hand-painted earthy accents. Strong material contrast appears across rough rock plates, worn ornamented ridges, and subtle decorative carved motifs embedded in the body. The image is rendered as hand-drawn 2D game concept art with crisp clean linework, painterly digital rendering, and expressive brush-driven strokes, designed as a readable 2D game asset with a clear silhouette. The gargoyle remains rooted in place, swaying dizzily from side to side in a loose off-balance posture, with its head wobbling, shoulders slumped, and wings twitching unevenly. Its clawed hands hang open and empty, and the beast-like stone body rocks in a compact loop that clearly conveys a stunned status effect without any forward movement.

Bottom row, left — HunyuanVideo-I2V + BiRefNet. An amorphous air elemental creature formed from swirling sky-blue, white, and pale cyan vapor masses appears as a readable elemental body silhouette built from soft cloudlike contours, curling wind ribbons, and floating wisps, accented with bright solid vector highlights and subtle ornamental airflow motifs. The subject is rendered as a clean 2D flat vector game asset in crisp SVG-

inspired styling, with sharp polished edges, minimal flat shading, clearly grouped color planes, and a palette of bright sky blue, white, and pale cyan enhanced by vibrant solid highlights for a stylized yet production-ready look. The view remains a strict top-down overhead angle, looking directly down from above in a 2D flat projection with no perspective, and the creature has no specific facing so its fluid amorphous form is shown naturally. In the looping animation, the air elemental stays in place while steadily dripping and leaking fluid beneath itself, with small repeated droplets forming along its lower vapor edges and falling straight downward in a compact cycle, as the main body gently quivers and ripples without shifting position.

Bottom row, right — HunyuanVideo-I2V + MatAnyone2. A sturdy wooden crate prop is shown from an isolated UI item showcase perspective at a clean neutral angle, facing in the most effective three-quarter direction to display its structure with maximum clarity. The crate has a clean boxy silhouette, reinforced corner joins, visible plank faces and slat seams, simple cross-brace paneling, and slightly beveled edges over solid flat surfaces, giving it a readable retro game shape. It is rendered in a 16-bit retro arcade pixel sprite style with pixel-perfect detailing, strong silhouette clarity, high-contrast flat 2D asset rendering, solid color blocking, and compact decorative wood grain marks formed by crisp pixel clusters. The color treatment uses warm medium brown, dark brown, and muted tan arranged in clean flat pixel masses, with deep brown accent pixels defining seams, edging, and depth separation. In the animation, the wooden crate performs a compact in-place hit reaction, giving a brief stagger and flinch as it snaps slightly backward with a tight jolt, while the corners and plank faces shift by a few pixels to suggest impact, then it eases cleanly back to its neutral resting pose.

D.4 Out-of-Domain RGBA Generation with Soft Transparency

We further examine the model's ability to generate RGBA videos with soft transparency from reference inputs beyond the GameAlpha training domain. Figure 18 presents representative frames from five generated RGBA videos, including wispy smoke, colored smoke, a luminous ring, and a planet-like object. The generated alpha maps capture fine smoke structures and gradual opacity transitions. Re-compositing the same generated foreground onto different backgrounds illustrates the model's ability to generate compositable RGBA videos with soft transparency across these content types.

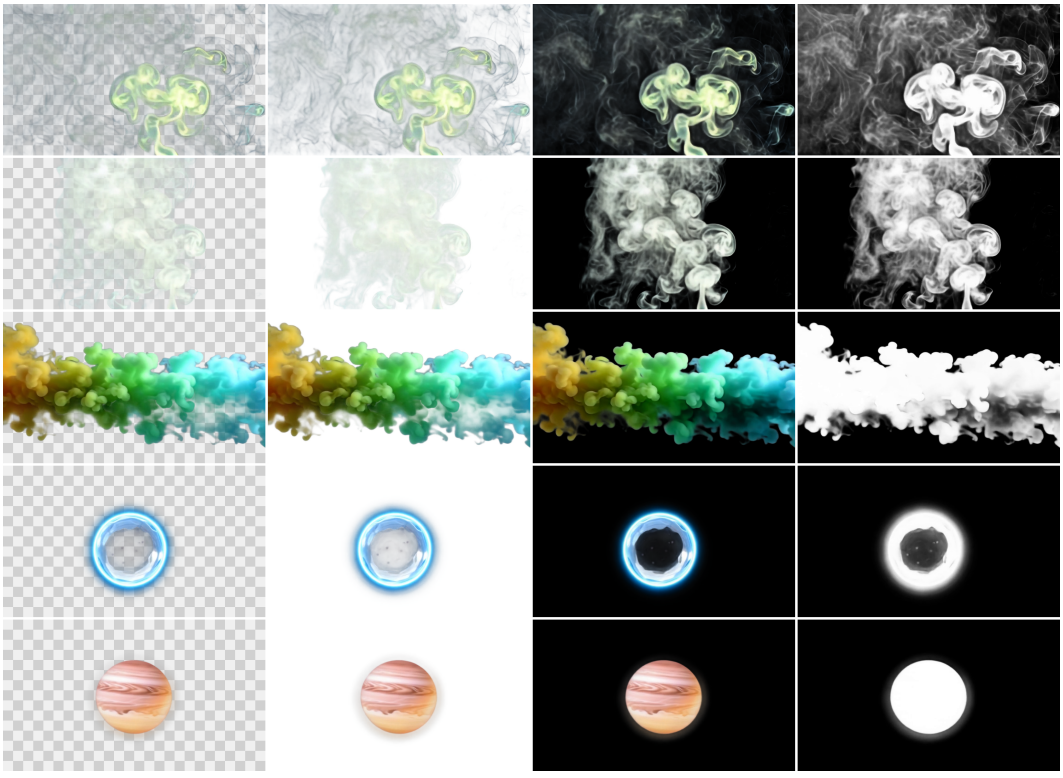


Figure 18: **Soft transparency in generated RGBA videos.** Each row shows the same frame from one generated video. Columns show composites on a gray-and-white checkerboard, white, and black backgrounds, followed by the corresponding generated alpha map.

D.5 Paired Out-of-Domain Visualizations

We complement the quantitative VideoMatte240K evaluation in Appendix C.7 with five qualitative examples in Figures 19 and 20. These human-subject inputs extend beyond the GameAlpha game-asset domain and cover varied appearances, accessories, objects, and poses. For each example, dense and visibility-routed inference use the same reference input, text prompt, and random seed, and we display the same frame index from both generated videos. The paired results show similar subject appearance, object structure, and alpha silhouettes across the two inference settings. Compositing each generated foreground onto checkerboard, white, and black backgrounds allows inspection of its appearance alongside the corresponding alpha map.

E Reproducibility Notes

GameAlpha-2.4K is split into 2,156 training and 239 validation clips before any training; the Stage-1 VAE trains on the training split plus VideoMatte240K, the Stage-2 DiT only on the training split, and the transparency probe is fitted on dense trajectories of training clips and frozen. The cross-domain diagnostic in Appendix C.7 is a stated exception: it uses a separately trained RGB-A VAE checkpoint that did not use VideoMatte240K. Unless otherwise stated, the GameAlpha automatic metrics in Sec. 4.1 use the full validation split ($n = 239$) at a fixed seed, except the three-seed evaluation of dense inference, Cache-KV, and matched no reactivation (Appendix C.5); auxiliary analyses, cross-domain experiments, and human evaluation use the sample sizes stated in their respective text or captions. We select $(k_{\text{route}}, r_s) = (2, 2)$ on GameAlpha training trajectories using the risk–yield analysis in Sec. 4.5 and keep this configuration fixed for validation and external evaluation. Further implementation details are intended for the planned dataset, code, and checkpoint releases; release status and licensing are described in Appendix F.

F Licenses and Intended Release

GameAlpha-2.4K and our implementation are not public at submission. We plan to release GameAlpha-2.4K, open-source the accompanying implementation, and release the trained model checkpoint. The artifact scope and applicable licenses will be specified at release.

Verified third-party terms. At submission, the official release pages report the following licenses; repository code, model weights, and datasets are distinguished because one artifact's license does not automatically cover the others.

- **Apache License 2.0.** Wan2.1 code and released models (Wan et al., 2025); LightX2V code and the four-step I2V distillation checkpoint used here (Contributors, 2025); standard DINOv2 code and weights (Oquab et al., 2024); VBench code (Huang et al., 2024); and UniVidX code and released checkpoints (Chen et al., 2026a).
- **MIT License.** The BiRefNet (Zheng et al., 2024) and Wan-Alpha (Dong et al., 2025) repository code and the LAION aesthetic-predictor repository use MIT. Wan-Alpha's released model cards separately label its weights Apache-2.0.
- **Custom or restricted licenses.** CogVideo code is Apache-2.0, but the CogVideoX1.5-5B-I2V parameters use the CogVideoX License (Yang et al., 2024). HunyuanVideo-I2V code and weights use the Tencent Hunyuan Community License Agreement (Kong et al., 2024). MatAnyone2 uses S-Lab License 1.0, which permits non-commercial use and requires permission for commercial use (Yang et al., 2026); SAM3 code and weights use the SAM License (Carion et al., 2025); and the RMBG-2.0 weights used with the BiRefNet architecture use CC BY-NC 4.0, with commercial use requiring a Bria agreement.
- **Data and metric caveats.** The official VideoMatte240K page (Lin et al., 2021a) does not specify a separate dataset license; we do not treat the associated repository's MIT software license as licensing the dataset. The StyleGAN-V I3D/FVD implementation used for FVD (Skorokhodov et al., 2021) has no standalone license file; its repository states that it is likely subject to the NVIDIA Source Code License-NC inherited through StyleGAN2-ADA.



Figure 19: **Paired out-of-domain RGBA generation on VideoMatte240K.** Each example contains two rows: dense inference above and visibility-routed inference below. Columns show composites on a gray-and-white checkerboard, white, and black backgrounds, followed by the generated alpha map.

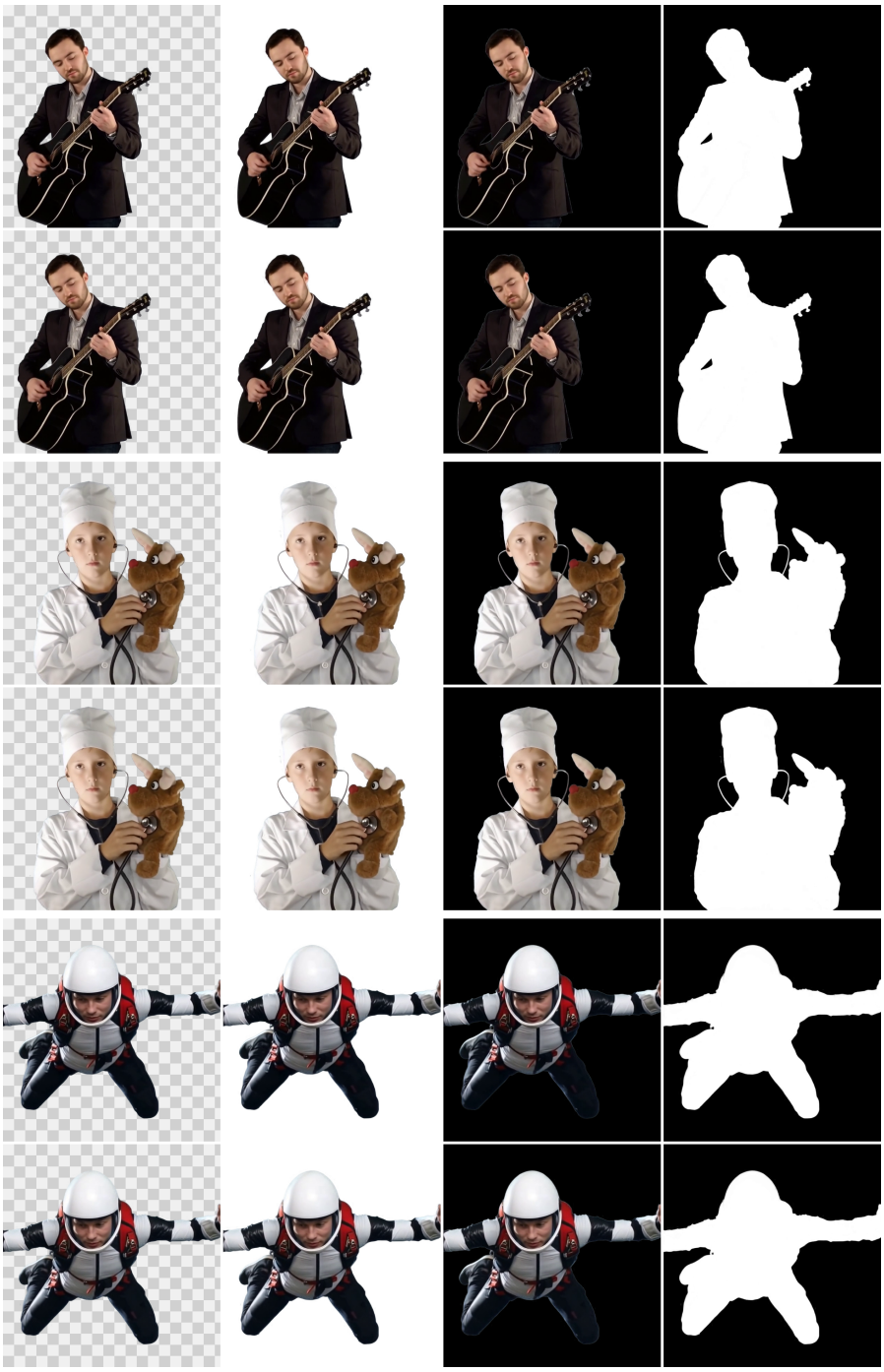


Figure 20: **Additional paired out-of-domain examples.** Guitar playing, a child holding a stuffed toy, and a skydiving pose. Row order and columns follow Figure 19.

All third-party resources, including bundled submodels and dependencies, remain governed by their upstream terms and are not relicensed by us. The planned release will reference upstream sources and record artifact-level licenses rather than redistribute third-party datasets or checkpoints unless their terms permit it. The fixed generation and screening services used for dataset construction are governed by their providers' service terms. Before releasing GameAlpha-2.4K, we will verify that those terms permit redistribution of the generated outputs.