# OMNIXREF: A UNIFIED FRAMEWORK FOR MULTIMODAL MULTI-REFERENCE VIDEO EDITING

**Anonymous authors**
Paper under double-blind review

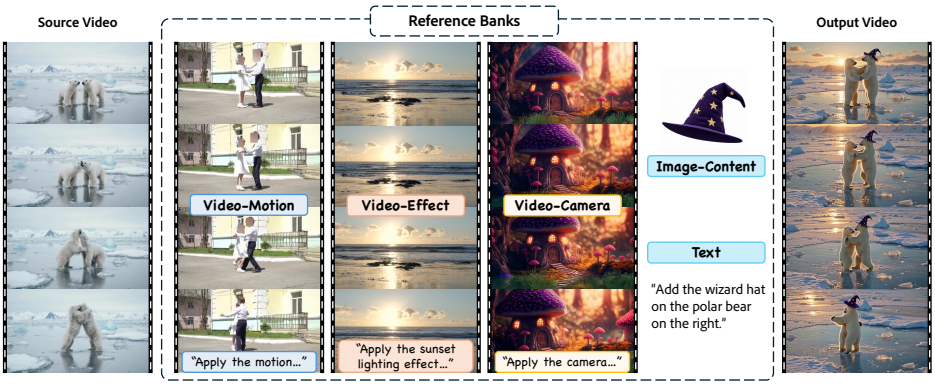



Figure 1: OmniXRef supports multimodal (text, image and video) multi-reference video editing, transferring the desired attribute from each reference to the source video.

## ABSTRACT

Creative intent in video editing can be expressed through text instructions (e.g., "remove the people"), image references (e.g., subject appearance), video references (e.g., camera movement), or a combination of the above. We introduce OmniXRef, a unified framework for multimodal multi-reference video editing that allows users to combine variable numbers of text, image, and video references to control diverse attributes of a source video. We first devise a unified input design with a reference-bank interface and introduce $R^2$-RoPE to support generalization to more references. Then our model, OmniXRef-Model, uses a single-stream diffusion transformer to jointly process the source video and multimodal context from an MLLM in an interleaved sequence. To address the scarcity of training data for video-reference editing, we construct OmniXRef-Data using synthesis pipelines for different editing tasks and combinations of references. We use curriculum learning to progressively increase task complexity and resolution. To support systematic evaluation of video as a control modality, we introduce OmniXRef-Bench, a benchmark for video-reference attribute transfer across Camera, Motion, Content, Style, and Visual Effects (VFX). Comprehensive evaluations show that our OmniXRef-Model outperforms task-specific baselines and commercial models.

## 1 INTRODUCTION

Video editing plays a central role in modern digital media creation, enabling applications from film production and advertising to personalized visual storytelling. While traditional video editing often requires time-consuming manual edits at each step of the editing pipeline, recent advances in generative models have made this more automatic. Given a source video, a video editing model aims to apply a specific edit guided by user intent (e.g., via a text prompt) while preserving everything else in the source video. However, describing intricate visual details and how they should dynamically change over time (e.g., camera, lighting, visual effects) can be highly challenging to specify through text and/or image guidance alone, leading to a need for a video reference. Going further, realizing a creative vision may require several multimodal references at once, each contributing a different aspect (e.g., object motion, visual effects, camera) of the intended result, as shown in Figure 1.



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.

Despite this need, reference-guided video editing has mostly been studied in settings constrained by a single reference modality. Existing methods support text instructions with an optional image reference (Cong et al., 2025; Mou et al., 2025; Ju et al., 2025; Lin et al., 2026b), multiple image references (Liu et al., 2026b), or a single video reference for a specific task such as camera control (Luo et al., 2025) and visual effects (VFX) (Jones et al., 2026). Frontier systems increasingly support broader combinations of multimodal references (ByteDance Seed, 2026b; Google DeepMind, 2026; MiniMax, 2026), though many only support references for generation (Kling AI, 2026; Alibaba Cloud Model Studio, 2026; Google Developers, 2025), not editing. In fact, a systematic formulation of *multimodal multi-reference video editing* remains largely underdeveloped.

To this end, we introduce *OmniXRef*, a unified framework for multimodal multi-reference video editing. Our omni-reference formulation supports four dimensions: (i) number of references (1, ..., $K$), (ii) reference modality (text, image, video), (iii) reference attribute (Camera, Motion, Content, Style, VFX) and (iv) composition (text-image, text-video, image-video, text-image-video). The reference inputs are organized through a common reference-bank interface, with each bank specifying an editing task and instruction, together with an image or video reference when available, allowing *OmniXRef-Model* to combine multiple requested edits into one coherent output. OmniXRef-Model integrates an MLLM (Bai et al., 2025b) with a single-stream diffusion transformer (DiT), where the MLLM first encodes an interleaved sequence of the system prompt, source video and reference banks to derive the semantic context, then the DiT jointly processes this semantic context, source and reference VAE tokens, and noisy target-video tokens to perform the requested edits while preserving the rest of the source content. To reduce the computational cost of conditioning on multiple video references, we introduce Mix-Patchify (MP), which adapts different patchification granularity for each reference according to its editing task. Furthermore, we introduce $R^2$-RoPE (RoPE with Reserved Reference Slots), which reserves a pool of positional slots for reference banks and randomly assigns banks to these slots during training, allowing support for more references at inference than seen in training. We train OmniXRef-Model with curriculum learning, progressively increasing task complexity and resolution to accelerate convergence.

Training OmniXRef-Model across reference modalities requires large-scale data covering different reference counts and diverse editing tasks. A key challenge is the limited availability of source–reference–target video triplets $(V_{\mathrm{src}}, V_{\mathrm{ref}}, V_{\mathrm{tgt}})$ covering a wide spectrum of editing tasks. Thus, we develop data synthesis pipelines for single-video-reference editing across five categories: Camera, Motion, Content, Style, and VFX, as well as a compositional data synthesis pipeline to support multi-reference editing. These are assembled with text-only and text-image data, resulting in *OmniXRef-Data* containing $\sim$10M training examples. Moreover, to address the absence of comprehensive video-reference editing benchmarks, we introduce *OmniXRef-Bench*, comprising 125 cases across the five categories. The evaluation protocol measures how faithfully models follow the requested edit and preserve unedited source content, along with the visual quality and temporal consistency of the output. Evaluations on OmniXRef-Bench show that OmniXRef-Model outperforms task-specific baselines in their respective domains and demonstrates better reference adherence and source preservation compared to strong commercial models such as Seedance 2.5 and MiniMax H3.

In summary, our contributions are fourfold:

- We introduce *OmniXRef*, a unified framework that formulates multimodal multi-reference video editing along four dimensions: reference number, modality, attribute, and composition. Our proposed reference-bank interface enables flexible combinations of references, with great potential for training at scale.
- We develop *OmniXRef-Model*, combining $R^2$-RoPE to support generalization to larger reference sets and Mix-Patchify to reduce the computational cost of conditioning on multiple video references.
- We construct *OmniXRef-Data* with data synthesis pipelines for each single-video-reference editing task as well as compositions of two visual references.
- We introduce *OmniXRef-Bench* to systematically evaluate reference adherence, visual quality, and source preservation, advancing future work towards video as a control modality for editing. Comprehensive evaluations show that our OmniXRef-Model outperforms task-specific baselines and commercial models.

## 2 Related Work

**Video Generation and Editing.** Video generation is evolving toward comprehensive content-creation systems for multimodal generation and editing (ByteDance Seed, 2026b; Google DeepMind, 2026; Kling Team, 2025; MiniMax, 2026). A central research direction is to expand the types of input that users can provide to specify an edit. In this paper, we denote each editing setting by its reference inputs: $\mathrm{T}'$, $\mathrm{I}'$, and $\mathrm{V}'$ indicate a text, image, and video reference, respectively, and $\mathrm{V2V}$ denotes source-to-output video editing. EditVerse (Ju et al., 2025), VIVA (Cong et al., 2025), InstructX (Mou et al., 2025), and Kiwi-Edit (Lin et al., 2026b) support $\mathrm{T}'\mathrm{V2V}$ and $\mathrm{I}'\mathrm{V2V}$. ReBind (Liu et al., 2026b) and TIDE (Liu et al., 2026a) support multiple reference images and associate each with its intended edit. Video references provide temporal information, including camera trajectories, dynamic lighting, and evolving visual effects (VFX), that image(s) cannot demonstrate and text alone may not specify precisely. CamCloneMaster (Luo et al., 2025) and RefVFX (Jones et al., 2026) study camera control and VFX transfer in the $\mathrm{V}'\mathrm{V2V}$ setting, respectively. Recent work (Zhang et al., 2026; Tan et al., 2025; Yang et al., 2026; Wang et al., 2026b; Qin et al., 2026; Wei et al., 2025; Chen et al., 2026; Pan et al., 2026; Wu et al., 2026) integrates multimodal understanding into video generation and editing through MLLM-based conditioning or joint modeling to handle complex compositions of instructions and references. Building on this progression, OmniXRef provides a unified formulation in which each text, image, or video reference is associated with an editing task and instruction, and multiple references can jointly specify the output.

**Video Editing Datasets and Benchmarks.** Training a general video editing model requires millions of high-quality data. Ditto-1M (Bai et al., 2025a) provides an open-sourced million-scale $\mathrm{T}'\mathrm{V2V}$ dataset. RefVIE (Lin et al., 2026b) and RefVideo-6M (Zi et al., 2026) are highlighted for large-scale $\mathrm{I}'\mathrm{V2V}$ data. Specialized $\mathrm{V}'\mathrm{V2V}$ datasets accompany camera and VFX transfer methods (Luo et al., 2025; Jones et al., 2026). However, large-scale source–reference–target triplets spanning diverse $\mathrm{V}'\mathrm{V2V}$ tasks and their compositions remain scarce. To address this gap, we construct OmniXRef-Data with data synthesis pipelines for each single-video-reference editing task as well as compositions of two visual references. For evaluation, ReCo-Bench (Zhang et al., 2025) and VEFX-Bench (Gao et al., 2026) evaluate $\mathrm{T}'\mathrm{V2V}$, EditVerseBench (Ju et al., 2025) and RefVIE-Bench (Lin et al., 2026b) cover $\mathrm{T}'\mathrm{V2V}$ and $\mathrm{I}'\mathrm{V2V}$, and IntelligentVBench (Pan et al., 2026), OmniVideo-1.0 (Kling Team, 2025), and TIDE-Bench (Liu et al., 2026a) extend evaluation to multimodal composition. However, none of them systematically evaluates transferring dynamic attributes from reference videos to a source video while preserving the remaining source content. OmniXRef-Bench fills this gap by systematically evaluating reference adherence, visual quality, and source preservation, advancing future work towards video as a control modality for editing.

## 3 OmniXRef-Model

### 3.1 Framework Overview

Coordinating multiple multimodal references within a single output requires a comprehensive understanding and reasoning of the whole input sequence. Hence, as shown in Figure 2, our OmniXRef-Model, $F_\theta$, first uses an MLLM ($G$), which is a frozen Qwen3-VL-8B-Instruct (Bai et al., 2025b) in practice, to encode an interleaved input sequence and derive its semantic context, then employs a pretrained single-stream video diffusion transformer (DiT), whose architecture is similar to EditVerse (Ju et al., 2025), to generate the edited video.

For a unified formulation, we represent text, image, and video references through a unified reference-bank interface. Each bank consists of a task definition, an edit instruction, and an optional image or video reference:

$$B_i = \big[\ \underbrace{T_i^{\mathrm{task}}}_{\text{Task definition}}\ ;\ \underbrace{T_i^{\mathrm{edit}}}_{\text{Edit instruction}}\ ;\ \underbrace{V_i}_{\text{Optional visual reference}}\ \big]. \tag{1}$$

For $i = 1, \ldots, K$, $T_i^{\mathrm{task}}$ defines the scope of the editing task, $T_i^{\mathrm{edit}}$ specifies the concrete edit instruction, and $V_i$ is the reference image or video. For a text-only bank, we set $T_i^{\mathrm{task}} = \emptyset$, $V_i = \emptyset$.

In OmniXRef-Model, a semantic context $\mathcal{X}$ is first obtained by:

$$\mathcal{X} = G(T_{\mathrm{sys}}, T_{\mathrm{src}}, \mathcal{E}_{\mathrm{ViT}}(V_{\mathrm{src}}), \{B_i\}_{i=1}^{K}, T_{\mathrm{out}}), \tag{2}$$

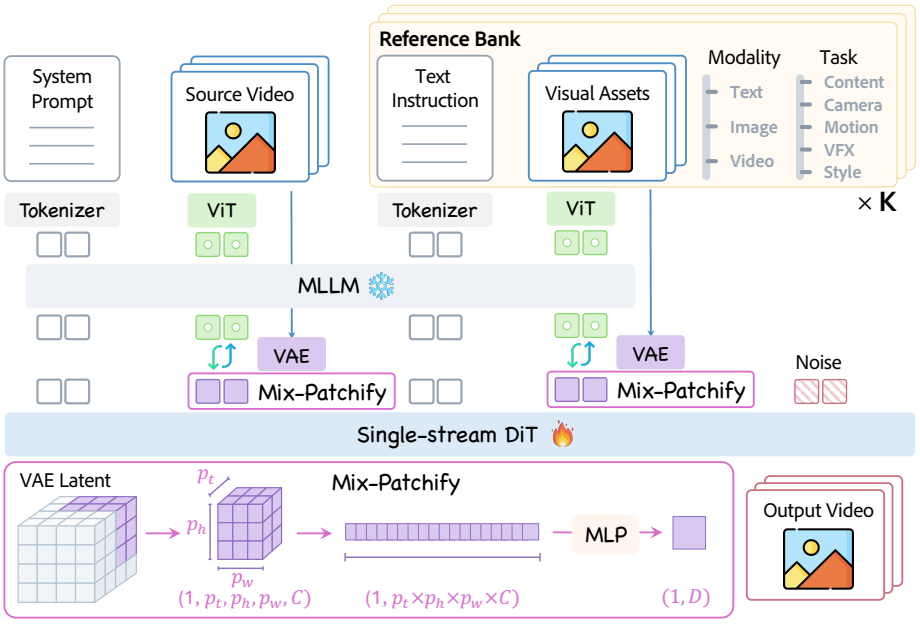



Figure 2: **Overview of OmniXRef-Model.** Reference banks provide a unified interface for text, image, and video references, enabling multimodal multi-reference video editing. An MLLM extracts semantic context from the interleaved inputs. A single-stream DiT jointly processes the contextual text tokens, visual tokens, and noise through full self-attention, allowing multiple reference banks to condition the edited output simultaneously.

where $T_{\text{sys}}$ is the system prompt that defines the overall reference-guided editing task, $T_{\text{src}}$ is the source marker *"This is the source video to edit:"*, $T_{\text{out}}$ is the output marker *"The output video is:"*, $V_{\text{src}}$ is the source video, and $\mathcal{E}_{\text{ViT}}$ is a ViT encoder. Note that $V_i$ in $B_i$ also goes through $\mathcal{E}_{\text{ViT}}$.

Then, the DiT, $H_\theta$, generates the output $H_\theta(\mathcal{X}', V_{\text{tgt}}; \theta)$, where $\mathcal{X}'$ is the semantic context after replacing all visual tokens in $\mathcal{X}$ to $\mathcal{E}_{\text{VAE}}(V)$, $\mathcal{E}_{\text{VAE}}$ is the VAE encoder, and $V_{\text{tgt}}$ is the ground truth edited video, which is noised during training. The DiT processes this sequence with full self-attention and denoises the noise tokens in context to generate the edited video.

### 3.2 Mix-Patchify

The patchification layer groups the VAE latents of a visual input into non-overlapping patches of size $p = (p_t, p_h, p_w)$ along the temporal, height, and width axes, and maps each flattened patch to a DiT token $\mathbf{e}_j$ with an MLP specific to $p$. The pretrained DiT uses $p = (1, 1, 1)$ for fine-grained denoising. We retain this granularity for the source and target tokens to preserve detailed source conditioning. Reference tokens, however, do not all require such fine spatial granularity. We therefore introduce *Mix-Patchify* (MP), which represents each reference at a spatial granularity suited to its editing task while keeping $p_t = 1$ throughout to preserve fine-grained temporal resolution for dynamic attributes. More details are provided in Appendix B.3. For a patch size $p$, MP reduces the token count by up to a factor of $p_t \cdot p_h \cdot p_w$. This substantially lowers training and inference costs (Appendix E), particularly with multiple references, while maintaining performance and facilitating convergence.

### 3.3 R$^2$-RoPE

Multimodal multi-reference video editing data become increasingly difficult to construct as the number of references increases, fundamentally restricting the model's ability to compose more references at inference. However, we attribute this limitation partly to a training–inference gap in the default sequential asset indexing of the 4D RoPE. By default, the 4D RoPE coordinate of token $\mathbf{e}_j$ is $\mathbf{q}_j = (y_j, x_j, \tau_j, \rho_s)$, where $(y_j, x_j, \tau_j)$ denote its height, width, and temporal coordinates, and $\rho_s$ is the sequential index of the asset to which the token belongs with regards to the full input sequence. Therefore, during training, the range of $\rho_s$ is bounded by the maximum number of assets in a training sequence. In such a case, when the number of references at inference exceeds those

seen during training, the model encounters out-of-distribution RoPE indices and relative offsets, hindering generalization.

To bypass this limitation, we propose $R^2$-RoPE (RoPE with Reserved Reference Slots), which modifies the fourth axis of the default 4D RoPE. Specifically, we reserve fixed slots for the system prompt, the source video, and the noise tokens, together with a dedicated pool of slots for reference banks, where the size of this pool is larger than the number of references supported with training data. All tokens belonging to the reference bank $B_i$ share the same fourth-axis coordinate $\rho_i$. The key is to randomly sample from the reserved reference slots during training. When a training example contains multiple references, non-overlapping slots are selected such that different banks receive distinct coordinates. As this design exposes every reserved reference slot to a reference input during training, additional references at inference time remain within the RoPE range seen during training, i.e., becomes *in-distribution*. $R^2$-RoPE thus avoids the training–inference gap and is able to support more number of references at inference time without requiring training examples for every reference number and combination. More details about $R^2$-RoPE are provided in Appendix B.4.

### 3.4 Curriculum Learning

We train OmniXRef-Model with a three-stage curriculum that progressively increases task complexity and output resolution: single-reference adaptation at 272P, multi-reference composition at 272P with two-visual-reference examples ($\mathrm{I'V'V2V}$ and $\mathrm{V'V'V2V}$), and high-resolution fine-tuning at 480P. This curriculum leads to faster and better convergence. Details are given in Appendix B.2.

## 4 Data

We assemble OmniXRef-Data to train OmniXRef-Model across reference modalities, numbers of references, editing tasks, and attribute combinations, resulting in ∼10M training examples spanning text-only ($\mathrm{T'V2V}$), one image or video reference ($\mathrm{(I/V)'V2V}$), and two visual references ($\mathrm{I'V'V2V}$ or $\mathrm{V'V'V2V}$).

**Text-only editing.** Our text-only data contains ∼7M high-quality in-house source–target pairs with diverse instructions. We retain this supervision throughout training to maintain general editing foundations, including instruction following and source video preservation. It also helps prevent the model from overfitting to a particular reference modality.

| **Task** | **#Samples** |
|---|---|
| $\mathrm{T'V2V}$ — *Text-only* | |
| General | 7M |
| $\mathrm{(I/V)'V2V}$ — *Single reference* | |
| Camera | 1.5M |
| Motion | 100K |
| Content | 650K |
| VFX | 100K |
| Style | 650K |
| $\mathrm{I'V'V2V}$ / $\mathrm{V'V'V2V}$ — *Two references* | |
| Content + Camera | 50K |
| Motion + Camera | 50K |
| **Total** | **10M** |

Table 1: Statistics of the training data.

**Visual-reference data.** We construct source–reference–target triplets $(V_{\mathrm{src}}, V_{\mathrm{ref}}, V_{\mathrm{tgt}})$ such that the source and target differ primarily in the edited attribute, while the reference and target share that attribute but depict a distinct content. This design discourages the model from directly copying reference content instead of transferring the intended attribute. We use Qwen3-VL-8B-Instruct (Bai et al., 2025b) as the MLLM throughout data construction. (1) For **VFX** and **Style**, which specify an evolving visual effect and a global appearance change such as stylization or dynamic lighting, respectively, we use a shared synthesis pipeline. The MLLM filters and captions a large-scale text-to-video collection, identifying ∼2M candidate target videos with effects such as flames, explosions, smoke, and particles, or with distinctive styles. For each target $V_{\mathrm{tgt}}$, GPT Image 2 (OpenAI, 2026) is used to remove the detected attribute (e.g., smoke) from its first frame using the instruction formulated by the MLLM. Next, a video propagation model (Liu et al., 2025) propagates this first-frame edit through $V_{\mathrm{tgt}}$ to produce $V_{\mathrm{src}}$, removing the attribute throughout the video while preserving its content. Then for $V_{\mathrm{ref}}$, MLLM describes a plausible different scene exhibiting the same attribute; GPT Image 2 generates the first frame using this prompt, and Wan2.2-TI2V-5B (Wan Team, 2025) generates a video from this first frame.

(2) For **Camera**, where the reference specifies a camera trajectory, framing, and timing, we sample two visually distinct videos, $V_1$ and $V_2$, from an in-house text-to-video collection. We reshoot both

with Vista4D (Lin et al., 2026a) under the same sampled camera path, obtaining $V_1^{\text{cam}}$ and $V_2^{\text{cam}}$ and thus two triplets: $(V_1, V_2^{\text{cam}}, V_1^{\text{cam}})$ and $(V_2, V_1^{\text{cam}}, V_2^{\text{cam}})$. The paths include primitives like orbits, dollies, cranes, optical zooms, pans and tilts, and freeform movements with varied scale, speed, framing, and rhythm.

(3) For **Motion**, where the reference specifies an object (e.g., human) pose sequence, we start from paired videos $(V_1, V_2)$ in MotionPair-60K (Yan et al., 2026), in which different characters perform the same motion. For each character, Wan2.2-TI2V-5B generates a video with different motion, yielding $V_1^{\text{alt}}$ and $V_2^{\text{alt}}$. We form $(V_1^{\text{alt}}, V_2, V_1)$ and $(V_2^{\text{alt}}, V_1, V_2)$, so that the reference demonstrates the target motion through a different character while the source retains the target character with a different motion. (4) For **Content**, we mix RefVIE (Lin et al., 2026b), which already provides image-reference triplets $(V_{\text{src}}, I_{\text{ref}}, V_{\text{tgt}})$ for inserting, removing, or replacing subjects and backgrounds, as well as video harmonization triplets in which a video reference supplies a dynamic background for a source foreground.

**Compositional references.** As the model cannot learn to *combine* multiple-reference attributes into one output only by learning from single-reference data, we obtain compositional data by applying the Camera reshooting pipeline to Content and Motion data pipelines, producing $\text{I}'\text{V}'\text{V2V}$ and $\text{V}'\text{V}'\text{V2V}$ data.

After MLLM quality filtering, the statistics of retained data are summarized in Table 1.

# 5 Experiments

## 5.1 Evaluation Setup

**Benchmarks.** To address the absence of comprehensive video-reference editing benchmarks, we introduce OmniXRef-Bench, which contains 125 samples across five categories: Camera, Motion, Content, Style, and VFX. Each sample pairs a source video with a reference video and specifies which attribute to transfer. We use Gemini 3.8 Flash (Google, 2026) as a VLM judge with a systematic evaluation prompt designed for each category; Details are deferred to Appendix C. Additionally, we also report ReCo-Bench (Zhang et al., 2025) and VEFX-Bench (Gao et al., 2026) to evaluate whether OmniXRef-Model preserves the foundational $\text{T}'\text{V2V}$ capability.

**Baselines.** Few existing methods support multimodal multi-reference video editing, so we mainly compare with the commercial models Seedance 2.0 (ByteDance Seed, 2026a), Seedance 2.5 (ByteDance Seed, 2026b), and MiniMax-H3 (MiniMax, 2026). We also compare with task-specific expert models: CamCloneMaster (Luo et al., 2025) for Camera, MotionFollower (Tu et al., 2025) for Motion, and RefVFX (Jones et al., 2026) for Content, Style, and VFX. On ReCo-Bench and VEFX-Bench, we compare with the methods on their leaderboards. In all tables, commercial systems are shown in grey for reference, and boldface marks the best result among the remaining methods.

**Metrics.** For OmniXRef-Bench, the VLM judge scores three aspects: Reference Adherence (RA), whether the output correctly transfers the requested attribute from the reference; Visual Quality (VQ), the visual and temporal quality of the output; and Source Preservation (SP), whether the remaining source content is preserved without importing unrelated reference content. Following CamI2V (Zheng et al., 2024), we additionally measure trajectory accuracy for the Camera reference, reporting Rotation Distance (RotErr), Translation Error (TransErr), and Camera Motion Consistency (CamMC). For ReCo-Bench and VEFX-Bench, we follow their official evaluation suites.

## 5.2 Quantitative Comparison

Table 2 reports the composite score for each category of OmniXRef-Bench, together with the five-category averages of RA, VQ, SP, and the composite score. OmniXRef-Model achieves the highest score in every category and the best overall average on RA and SP, outperforming both task-specific baselines and commercial systems. We attribute the VQ gap to commercial systems mainly to the capacity of the pretrained backbone. Nevertheless, our proposed model designs allow OmniXRef-Model to achieve state-of-the-art results on OmniXRef-Bench, and we expect OmniXRef to deliver further gains as stronger pretrained video models become available. Geometry-based camera tra-

Table 2: VLM evaluation of video-reference attribute transfer on OmniXRef-Bench. We report the composite score for each category, together with Reference Adherence (RA), Visual Quality (VQ), Source Preservation (SP), and the overall average (Score) over the five categories. All scores are the higher the better. Per-category RA, VQ, and SP are reported in Table 10.

| Model | Score by category | | | | | Overall avg. | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Camera | Motion | Content | Style | VFX | RA | VQ | SP | Score |
| OPEN-SOURCE | | | | | | | | | |
| CamCloneMaster | 3.20 | – | – | – | – | – | – | – | – |
| MotionFollower | – | 2.14 | – | – | – | – | – | – | – |
| RefVFX | – | – | 1.98 | 2.24 | 2.49 | – | – | – | – |
| OURS | | | | | | | | | |
| **OmniXRef** | **3.81** | **3.27** | **4.48** | **3.74** | **3.55** | **3.66** | **3.67** | **4.69** | **3.77** |
| COMMERCIAL | | | | | | | | | |
| MiniMax-H3 | 2.07 | 0.69 | 2.23 | 0.94 | 1.32 | 2.46 | 3.84 | 2.30 | 1.45 |
| Seedance 2.0 | 3.27 | 2.45 | 3.73 | 3.22 | 2.92 | 3.32 | 3.83 | 4.17 | 3.12 |
| Seedance 2.5 | 3.07 | 2.00 | 4.04 | 2.30 | 2.22 | 3.21 | 3.70 | 3.71 | 2.73 |

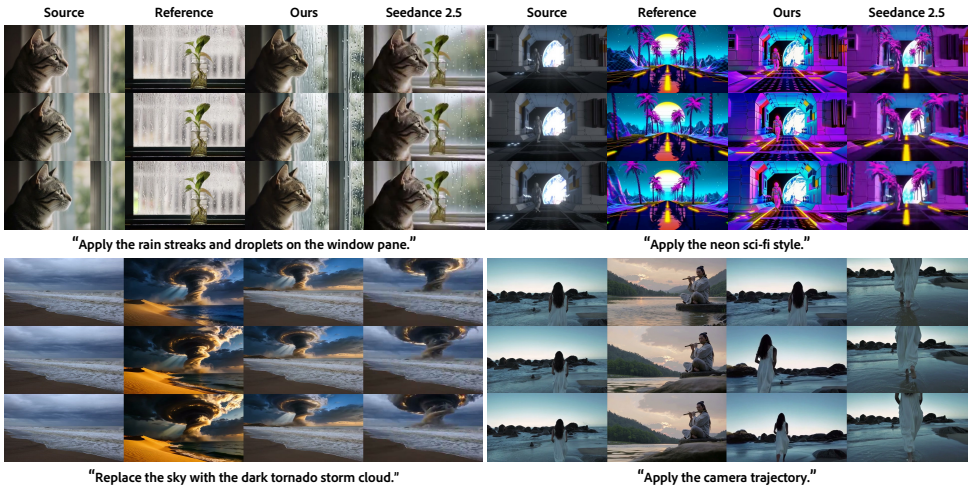



Figure 3: Qualitative comparison of OmniXRef-Model and Seedance 2.5 on V′V2V across visual effects, style, content, and camera editing.

jectory errors further confirm the strong camera adherence of OmniXRef-Model, and results on VEFX-Bench and ReCo-Bench show that it retains competitive T′V2V capability (Appendix D).

### 5.3 QUALITATIVE COMPARISON

Figure 3 compares OmniXRef-Model with Seedance 2.5 on four V′V2V cases spanning VFX, Style, Content, and Camera. Consistent with Table 2, OmniXRef-Model shows clear advantages in reference adherence and source preservation. For example, in the top-left sample in Figure 3, Seedance 2.5 responds to the rain but does not reproduce the droplets shown in the reference. It also alters the window frame, failing to preserve unedited source content, and even imports the potted plant from the reference, a form of reference leakage that also occurs in the top-right sample. In contrast, OmniXRef-Model precisely transfers the reference rain streaks and droplets onto the window pane while keeping the rest of the scene unchanged and avoiding reference leakage.

### 5.4 ABLATION STUDIES

We ablate the key designs of OmniXRef-Model, including Mix-Patchify (Section 3.2), $R^2$-RoPE (Section 3.3), curriculum learning (Section 3.4), and the dropout strategy, and report the results in Table 3, with the per-category breakdown in Table 11.

Table 3: Ablations of OmniXRef-Model on OmniXRef-Bench. Columns follow Table 2; per-category RA, VQ, and SP are reported in Table 11.

| Variant | Score by category | | | | | Overall avg. | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Camera | Motion | Content | Style | VFX | RA | VQ | SP | Score |
| w/o Mix-Patchify | 3.24 | 2.11 | 2.97 | 2.71 | 3.30 | 3.16 | **3.67** | 3.82 | 2.86 |
| Reference-bank dropout | 2.85 | 2.11 | 2.48 | 3.29 | 3.42 | 3.01 | 3.58 | 3.91 | 2.83 |
| w/o $R^2$-RoPE | 3.61 | 2.44 | 3.06 | 3.14 | **3.64** | 2.86 | 3.60 | 4.50 | 3.18 |
| w/o curriculum learning | 3.32 | 2.57 | 3.07 | 3.15 | 3.51 | 2.99 | 3.59 | 4.35 | 3.13 |
| **OmniXRef (full)** | **3.81** | **3.27** | **4.48** | **3.74** | 3.55 | **3.66** | **3.67** | **4.69** | **3.77** |

**Mix-Patchify.** Although applying $(1, 1, 1)$ patches to all references preserves the most complete reference information, it provides no spatial prior: the model must learn on its own how to extract the requested attribute from long, information-dense reference sequences, which makes learning harder and slows convergence within the same number of training steps. In contrast, Mix-Patchify uses task-specific patch sizes as a spatial prior, keeping fine-grained patches where appearance details matter (Content) and coarser patches where the attribute is mainly global or dynamic (Camera, Motion, Style, and VFX), which reduces the learning difficulty, making the model converge faster.

**$R^2$-RoPE.** Replacing $R^2$-RoPE with the default sequential asset indexing causes the smallest overall degradation among the ablations. We attribute this degradation to an intrinsic indexing conflict: under sequential indexing, the fourth-axis index of the noise tokens shifts with the number of references, so for example, the same index denotes the output in samples with one reference but a reference in samples with more. Mixing such samples during training therefore introduces conflicting positional signals that make learning harder.

**Curriculum learning.** Without curriculum learning, the model is trained at 480P on the full data pool. Within the same number of training steps, it must acquire all capabilities at once without the inductive bias of easy-to-hard training, which hinders optimization and degrades performance. Training at 480P throughout also makes each step considerably slower.

**Reference-bank dropout.** By default, we drop only the editing instruction $T_i^{\text{edit}}$ for classifier-free guidance (Ho & Salimans, 2022), so that the unconditional prediction remains grounded in the same visual reference and the guidance amplifies only the attribute extraction specified by the text instruction. In contrast, dropping the entire bank makes the guidance amplify the influence of the whole reference rather than only the requested attribute, which likely induces reference leakage and degrades performance.

### 5.5 Analysis of Reference Conditioning

**Responsiveness to visual references.** To verify that OmniXRef-Model actually uses the visual asset in the reference bank rather than relying on the editing instruction alone, we fix the source video and the instruction and only change the reference video. In Figure 4, two references with different sunset lighting are paired with the same instruction, "Apply the sunset...". The two outputs reproduce the distinct lighting of their respective visual references, confirming that the visual asset determines the appearance while the instruction specifies which attribute to transfer.

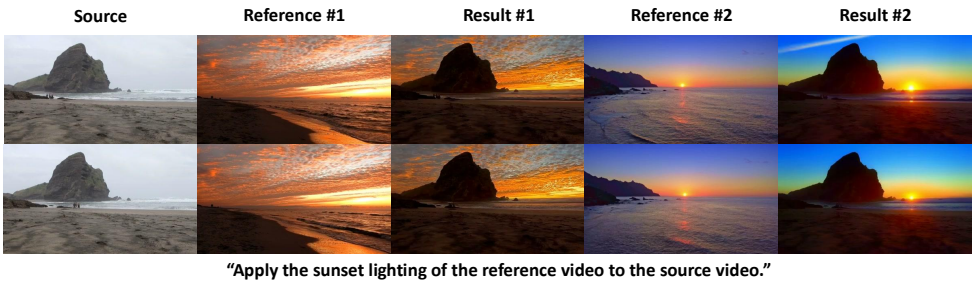



Figure 4: Responsiveness to different visual references with a fixed source video and edit instruction.

**Robustness to reference order.** With multiple references, a natural question is whether the order of the reference banks, that is, which $R^2$-RoPE slot each bank occupies, affects the result. To test this, we keep everything the same while swapping the two reference banks between their slots. As shown in Figure 5, both orders correctly transfer the camera motion and replace the table, indicating that OmniXRef-Model does not collapse into associating particular slots with particular attributes. This robustness stems from randomized slot assignment, which exposes every reserved slot to diverse references during training. As a result, although each training sample contains at most two visual references, all reserved slots are activated without bias.

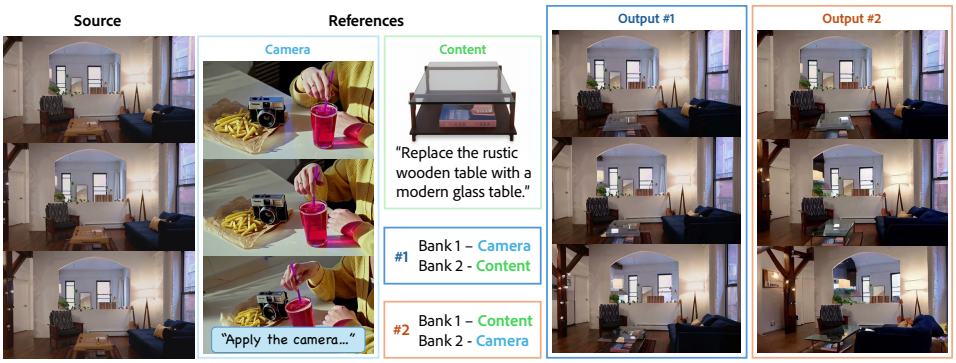



Figure 5: Reference-order robustness when swapping reference bank slots in $R^2$-RoPE.

**Generalization to more references.** Figure 6 composes three video references specifying subject motion, camera trajectory, and style, exceeding the maximum of two visual references in any training sample. OmniXRef-Model follows all three references while preserving the source subject and scene thanks to the $R^2$-RoPE design, whereas Seedance 2.5 largely fails to follow the motion and camera references and only partially reproduces the requested style.

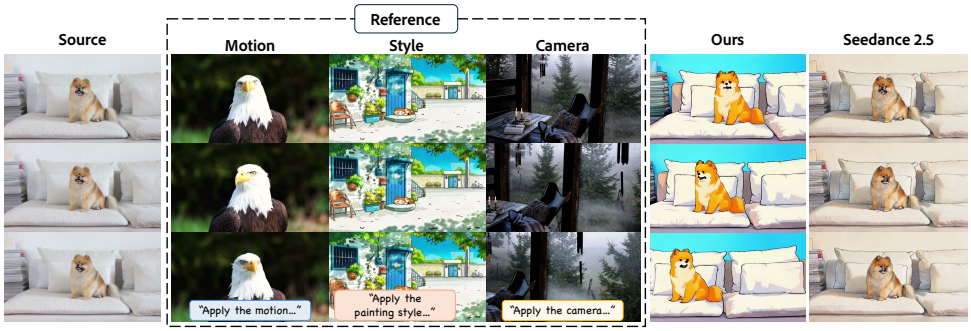



Figure 6: Compositional generalization to three video references.

## 6 Conclusion

We introduced OmniXRef, a unified framework for multimodal multi-reference video editing, in which text, image, and video references are organized through a common reference-bank interface and composed in variable numbers to control distinct attributes of a source video. OmniXRef-Model integrates an MLLM with a single-stream DiT, employs Mix-Patchify to enable efficient conditioning on multiple video references, and adopts $R^2$-RoPE to generalize beyond the number of references observed during training. To address the scarcity of video-reference editing data and benchmarks, we constructed OmniXRef-Data through synthesis pipelines for Camera, Motion, Content, Style, and VFX, as well as compositions of two references, and established OmniXRef-Bench for the systematic evaluation of reference adherence, visual quality, and source preservation. Extensive experiments demonstrate that OmniXRef-Model outperforms both task-specific baselines and commercial systems on OmniXRef-Bench. Further analyses verify that OmniXRef-Model exploits the visual content of each reference, remains robust to the ordering of references, and generalizes to reference compositions beyond the training distribution.

## AI USE STATEMENT

In this work, we used generative AI tools to maintain the codebase and polish the manuscript for clarity and readability. Generative AI models were used to synthesize and filter training data in our data generation pipeline as described in Sec. 4. A VLM judge was used to evaluate the results as described in Sec. 5. We did not use generative AI tools for brainstorming ideas, designing our method, or interpreting the experimental results. The authors reviewed all AI-assisted edits to the manuscript and verified and tested all AI-assisted code for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Alibaba Cloud. Alibaba Unveils Wan2.6 Series Enabling Everyone to Star in Videos. https://www.alibabacloud.com/blog/alibaba-unveils-wan2-6-series-enabling-everyone-to-star-in-videos_602742, 2025. Published December 16, 2025. Accessed September 25, 2026.

Alibaba Cloud Model Studio. HappyHorse: Reference-to-Video API Reference. https://docs.modelstudio.console.alibabacloud.com/en/model-studio/happyhorse-reference-to-video-api-reference, 2026. Official documentation. Accessed September 25, 2026.

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling Instruction-Based Video Editing with a High-Quality Synthetic Dataset, 2025a. URL https://arxiv.org/abs/2510.15742.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report, 2025b. URL https://arxiv.org/abs/2511.21631.

ByteDance Seed. Seedance 2.0 Official Launch. https://seed.bytedance.com/en/blog/seedance-2-0-official-launch, 2026a. Published February 12, 2026. Accessed September 21, 2026.

ByteDance Seed. One-take Creation, Flexible Referencing: Introducing Seedance 2.5. https://seed.bytedance.com/en/blog/one-take-creation-flexible-referencing-introducing-seedance-2-5, 2026b. Published July 31, 2026. Accessed September 17, 2026.

Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. VINO: A Unified Visual Generator with Interleaved OmniModal Context, 2026. URL https://arxiv.org/abs/2601.02358.

Xiaoyan Cong, Haotian Yang, Angtian Wang, Yizhi Wang, Yiding Yang, Canyu Zhang, and Chongyang Ma. VIVA: VLM-Guided Instruction-Based Video Editing with Reward Optimization, 2025. URL https://arxiv.org/abs/2512.16906.

Xiangbo Gao, Sicong Jiang, Bangya Liu, Xinghao Chen, Minglai Yang, Siyuan Yang, Mingyang Wu, Jiongze Yu, Qi Zheng, Haozhi Wang, Jiayi Zhang, Jie Yang, Zihan Wang, Qing Yin, and Zhengzhong Tu. VEFX-Bench: A Holistic Benchmark for Generic Video Editing and Visual Effects, 2026. URL https://arxiv.org/abs/2604.16272.

Google. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber. https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/, 2026. Published September 2, 2026. Accessed September 25, 2026.

Google DeepMind. Gemini Omni. https://deepmind.google/models/gemini-omni/, 2026. Official model page. Accessed September 17, 2026.

Google Developers. Introducing Veo 3.1 and New Creative Capabilities in the Gemini API. https://developers.googleblog.com/introducing-veo-3-1-and-new-creative-capabilities-in-the-gemini-api/, 2025. Published October 15, 2025. Accessed September 25, 2026.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-One Video Creation and Editing, 2025. URL https://arxiv.org/abs/2503.07598.

Maxwell Jones, Rameen Abdal, Or Patashnik, Ruslan Salakhutdinov, Sergey Tulyakov, Jun-Yan Zhu, and Kuan-Chieh Jackson Wang. Tuning-free Visual Effect Transfer across Videos, 2026. URL https://arxiv.org/abs/2601.07833.

Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, Daniil Pakhomov, Zhe Lin, Soo Ye Kim, and Qiang Xu. EditVerse: Unifying Image and Video Editing and Generation with In-Context Learning, 2025. URL https://arxiv.org/abs/2509.20360.

Kling AI. Kling VIDEO 3.0 Model User Guide. https://app.klingai.com/cn/quickstart/klingai-video-3-model-user-guide, 2026. Published February 6, 2026. Accessed September 25, 2026.

Kling Team. Kling-Omni Technical Report, 2025. URL https://arxiv.org/abs/2512.16776.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video Reshooting with 4D Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32671–32682, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_Vista4D_Video_Reshooting_with_4D_Point_Clouds_CVPR_2026_paper.html.

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-Edit: Versatile Video Editing via Instruction and Reference Guidance, 2026b. URL https://arxiv.org/abs/2603.02175.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, 2022. URL https://arxiv.org/abs/2210.02747.

Qi Liu, Gang Yue, Mingyu Yin, Lisai Zhang, Yidi Wu, Yaole Wang, Yaohui Wang, Chang Yao, Jingyuan Chen, and Lin Ma. TIDE: Task-Isolated Diffusion for Unified Video Editing and Generation, 2026a. URL https://arxiv.org/abs/2606.08260.

Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, and Jiaya Jia. Generative video propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2412.19761.

Xinyu Liu, Shihao Li, Weihong Lin, Xinlong Chen, Yang Shi, Yujin Han, Yiyang Cai, Yanghao Wang, Ruibin Yuan, Yuanxing Zhang, Pengfei Wan, Wenhan Luo, and Yike Guo. ReBind: Multi-Reference Video Editing via Structured Instructions with Explicit Reference Relationships, 2026b. URL https://arxiv.org/abs/2607.14681.

Yawen Luo, Jianhong Bai, Xiaoyu Shi, Menghan Xia, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Tianfan Xue. CamCloneMaster: Enabling Reference-based Camera Control for Video Generation, 2025. URL https://arxiv.org/abs/2506.03140.

MiniMax. MiniMax-H3. https://huggingface.co/MiniMaxAI/MiniMax-H3, 2026. Official model card and reference prompting guide. Accessed September 17, 2026.

Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. InstructX: Towards Unified Visual Editing with MLLM Guidance, 2025. URL https://arxiv.org/abs/2510.08485.

OpenAI. GPT Image 2 Model. https://developers.openai.com/api/docs/models/gpt-image-2, 2026. Official model documentation.

Kaihang Pan, Qi Tian, Jianwei Zhang, Weijie Kong, Jiangfeng Xiong, Yanxin Long, Shixue Zhang, Haiyi Qiu, Tan Wang, Zheqi Lv, Yue Wu, Liefeng Bo, Siliang Tang, and Zhao Zhong. OmniWeaving: Towards Unified Video Generation with Free-form Composition and Reasoning, 2026. URL https://arxiv.org/abs/2603.24458.

Luozheng Qin, Jia Gong, Qian Qiao, Tianjiao Li, Li Xu, Haoyu Pan, Chao Qu, Zhiyu Tan, and Hao Li. Uni-ViGU: Towards Unified Video Generation and Understanding via A Diffusion-Based Video Generator, 2026. URL https://arxiv.org/abs/2604.08121.

Zhiyu Tan, Hao Yang, Luozheng Qin, Jia Gong, Mengping Yang, and Hao Li. Omni-Video: Democratizing Unified Video Understanding and Generation, 2025. URL https://arxiv.org/abs/2507.06119.

Shuyuan Tu, Qi Dai, Zihao Zhang, Sicheng Xie, Zhi-Qi Cheng, Chong Luo, Xintong Han, Zuxuan Wu, and Yu-Gang Jiang. MotionFollower: Editing Video Motion via Score-Guided Diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12822–12831, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Tu_MotionFollower_Editing_Video_Motion_via_Score-Guided_Diffusion_ICCV_2025_paper.html.

Wan Team. Wan2.2: Text-Image-to-Video 5B. https://github.com/Wan-Video/Wan2.2, 2025. Official model repository.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-$\omega$. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://arxiv.org/abs/2605.15195.

Yuqi Wang, Runyi Li, Ruoyu Feng, Renjie Chen, Wenfeng Lin, and Mingyu Guo. Bridging Video Understanding and Generation in a Unified Framework, 2026b. URL https://arxiv.org/abs/2606.31326.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. UniVideo: Unified Understanding, Generation, and Editing for Videos, 2025. URL https://arxiv.org/abs/2510.08377.

Jianzong Wu, Hao Lian, Jiongfan Yang, Dachao Hao, Ye Tian, Yunhai Tong, Jingyuan Zhu, Biaolong Chen, Qiaosong Qi, Aixi Zhang, Wanggui He, Mushui Liu, Jinlong Liu, Pipei Huang, and Hao Jiang. LoomVideo: Unifying Multimodal Inputs into Video Generation and Editing, 2026. URL https://arxiv.org/abs/2606.06042.

xAI. Grok Imagine API. https://x.ai/news/grok-imagine-api, 2026. Published January 28, 2026. Accessed September 25, 2026.

Wenhao Yan, Fengjia Guo, Zhuoyi Yang, and Jie Tang. SCAIL-2: Unifying Controlled Character Animation with End-to-end In-Context Conditioning, 2026. URL https://arxiv.org/abs/2606.10804.

Hao Yang, Zhiyu Tan, Jia Gong, Luozheng Qin, Hesen Chen, Xiaomeng Yang, Yuqing Sun, Yuetan Lin, Mengping Yang, and Hao Li. Omni-Video 2: Scaling MLLM-Conditioned Diffusion for Unified Video Generation and Editing, 2026. URL https://arxiv.org/abs/2602.08820.

Pengze Zhang, Yanze Wu, Mengtian Li, Xu Bai, Songtao Zhao, Fulong Ye, Chong Mou, Xinghui Li, Zhuowei Chen, Qian He, and Mingyuan Gao. OmniTransfer: All-in-one Framework for Spatio-temporal Video Transfer, 2026. URL https://arxiv.org/abs/2601.14250.

Zhongwei Zhang, Fuchen Long, Wei Li, Zhaofan Qiu, Wu Liu, Ting Yao, and Tao Mei. Region-Constraint In-Context Generation for Instructional Video Editing, 2025. URL https://arxiv.org/abs/2512.17650.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. CamI2V: Camera-Controlled Image-to-Video Diffusion Model, 2024. URL https://arxiv.org/abs/2410.15957.

Bojia Zi, Xiaoyan Yang, Yu Zhou, Ruijie Sun, Lihan Zhang, Bin Liang, Kam-Fai Wong, Haibin Huang, Chi Zhang, and Xuelong Li. RefVideo-6M: A Reliable Reference-Based Dataset for Instructional Video Editing, 2026. URL https://arxiv.org/abs/2608.26101.

# A Interleaved Sequence Specification

## A.1 System Prompt

The system prompt $T_{\text{sys}}$ is shared by all reference configurations:

> You receive a source video, then one or more edit banks, each containing a task definition, an editing instruction, and optionally its own reference image or video. Apply each bank using only its own reference, take nothing else from any reference, and keep everything the banks do not change as in the source. Output the edited video.

## A.2 Source and Output Markers

Two fixed text markers delimit the roles of the videos in the sequence. The source marker $T_{\text{src}}$, `This is the source video to edit:`, immediately precedes the source video and identifies it as the video whose unedited content must be preserved. The output marker $T_{\text{out}}$, `The output video is:`, immediately precedes the target tokens and signals where the edited video is generated.

## A.3 Reference Banks

Each bank $B_i$ is a self-contained edit clause consisting of three parts.

**Task definition.** The task definition $T_i^{\text{task}}$ fixes the category of the bank and restricts what may be taken from its reference, while requiring the remaining source content to be preserved. It thereby specifies *which attribute* to extract, independently of the free-form text instruction. Table 4 lists the task definition of each category verbatim.

Table 4: Task definition $T_i^{\text{task}}$ of each category

| Category | Task definition |
|---|---|
| Camera | "This is a camera edit — change only the camera trajectory, keep everything else exactly as in the source." |
| Motion | "This is a motion edit — change only the motion, keep everything else exactly as in the source." |
| Content | "This is a content edit — change only the indicated subject or background, keep everything else exactly as in the source." |
| VFX | "This is a visual effect edit — add only the indicated phenomenon, keep everything already in the source unchanged." |
| Style | "This is a style edit — change only the lighting or stylization, keep everything else exactly as in the source." |

**Editing instruction.** The editing instruction $T_i^{\text{edit}}$ is a free-form description of the concrete edit, for example, "Replace the sky with the dark tornado storm cloud."

**Visual reference.** The optional visual reference $V_i$ is an image or a video that provides the content of the requested attribute.

For a text-only bank, $T_i^{\text{task}} = \emptyset$ and $V_i = \emptyset$, so the bank contains only the editing instruction. When an example contains several banks, they are concatenated after the source video, each with its own task definition, instruction, and reference. The same layout therefore covers text-only editing (T′V2V), single visual references (I′V2V and V′V2V), and their compositions (e.g., I′V′V2V and V′V′V2V) without any change to the model.

# B Implementation Details

## B.1 Training Objective

Let $Z_0 = \mathcal{E}_{\text{VAE}}(V_{\text{tgt}})$ be the clean target VAE latent. At noise level $t \in [0, 1]$, with $\epsilon \sim \mathcal{N}(0, I)$, we construct the noisy latent $Z_t = (1 - t)Z_0 + t\epsilon$. The DiT processes the interleaved sequence $\mathcal{X}'$, in which $Z_t$ occupies the target positions, and predicts the velocity $u_\theta(Z_t, t)$ at these target positions. It is optimized with the flow-matching loss (Lipman et al., 2022):

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{Z_0,t,\epsilon}\|u_\theta(Z_t, t) - (Z_0 - \epsilon)\|_2^2 . \tag{3}$$

## B.2 Curriculum Learning

To fully exploit the understanding capability of the MLLM and the generative capability of the DiT, we train OmniXRef-Model with 32 NVIDIA H100 GPUs in three stages that progressively increase task complexity and output resolution, leading to faster and better convergence.

Table 5: Training data of OmniXRef-Data with the sampling weight of each task at each curriculum stage.

| **Task** | **Weight** | | **#Samples** |
|---|---|---|---|
| | Stage 1 | Stage 2/3 | |
| T′V2V — *Text-only* | | | |
| General | 20% | 20% | 7M |
| (I/V)′V2V — *Single-visual-reference* | | | |
| Camera | 16% | 8% | 1.5M |
| Motion | 24% | 12% | 100K |
| Content | 16% | 8% | 650K |
| VFX | 16% | 8% | 100K |
| Style | 8% | 4% | 650K |
| I′V′V2V / V′V′V2V — *Compositional references* | | | |
| Content + Camera | 0% | 20% | 50K |
| Motion + Camera | 0% | 20% | 50K |
| **Total** | **100%** | **100%** | **10M** |

**Stage 1: Single-reference adaptation.** We begin at 272P resolution with a mixture of T′V2V, I′V2V, and V′V2V examples. This stage adapts the pretrained model to the unified reference-bank interface and establishes task-specific transfer ability. The reduced resolution lowers computational cost and accelerates convergence. The learning rate in this stage starts at $1.5 \times 10^{-5}$ and decays to $1 \times 10^{-5}$ over the first 5,000 steps.

**Stage 2: Multi-reference composition.** Keeping the 272P resolution, we retain the Stage 1 tasks and introduce I′V′V2V and V′V′V2V examples, each composing two visual references. In this stage, the model learns to jointly use multiple $R^2$-RoPE slots to combine references across modalities, laying the foundation for generalization to more reference compositions within the reserved capacity. The learning rate is kept at $1 \times 10^{-5}$.

**Stage 3: High-resolution fine-tuning.** We increase the resolution of all training examples to 480P while keeping all other Stage 2 settings unchanged.

## B.3 Mix-Patchify

The patchification layer converts continuous VAE latents into DiT tokens. For a visual input $V$, the VAE encoder $\mathcal{E}_{\text{VAE}}$ produces $Z = \mathcal{E}_{\text{VAE}}(V) \in \mathbb{R}^{L\times H\times W\times C}$, where $L$, $H$, $W$, and $C$ denote the number of latent frames, latent height, latent width, and latent channel dimension, respectively. A

patch size $p = (p_t, p_h, p_w)$ specifies the number of latents grouped along the temporal, height, and width axes, respectively. We partition $Z$ into $N_p$ non-overlapping patches, with zero-padding at the boundaries, where

$$N_p = \left\lceil \frac{L}{p_t} \right\rceil \left\lceil \frac{H}{p_h} \right\rceil \left\lceil \frac{W}{p_w} \right\rceil. \tag{4}$$

For $j = 1, \ldots, N_p$, let $\mathcal{P}_j$ denote the set of latent positions in the $j$-th patch, which contains $r = p_t \cdot p_h \cdot p_w$ latents. The corresponding patch $Z_{\mathcal{P}_j}$ is flattened and mapped to a DiT token $\mathbf{e}_j$ by an MLP $\mathbb{P}_p$ specific to patch size $p$:

$$\mathbf{e}_j = \mathbb{P}_p(\mathrm{vec}(Z_{\mathcal{P}_j})) \in \mathbb{R}^D, \tag{5}$$

where $D$ is the DiT token dimension and $\mathrm{vec}(Z_{\mathcal{P}_j}) \in \mathbb{R}^{rC}$ is the channel-wise concatenation of the $r$ latents in $Z_{\mathcal{P}_j}$. Each visual input $V$ is thus represented by $N_p$ DiT tokens. The pretrained DiT uses $p = (1, 1, 1)$ for fine-grained denoising. We retain this granularity for the source and target tokens to preserve detailed source conditioning. Reference tokens, however, do not all require such fine spatial granularity. We therefore introduce ***Mix-Patchify*** (MP), which represents each reference at a spatial granularity suited to its editing task while keeping $p_t = 1$ throughout to preserve fine-grained temporal resolution for dynamic attributes. For a patch size $p$, MP reduces the token count by up to a factor of $p_t \cdot p_h \cdot p_w$. This substantially lowers training and inference costs, particularly with multiple references, while maintaining performance and facilitating convergence.

In practice, the task category of each reference bank determines its patch size: Content references retain $(1, 1, 1)$ patches to preserve identity and fine appearance details, while Camera, Motion, Style, and VFX references use spatially coarser $(1, 2, 2)$ patches. To initialize a coarse-patch MLP $\mathbb{P}_p$ from the pretrained model, we set it to produce the average of the outputs of the pretrained fine-grained MLP $\mathbb{P}_{(1,1,1)}$ over the $r$ latents in each patch $Z_{\mathcal{P}_j}$, and then optimize it without this constraint. To maintain a shared RoPE space across granularities, each coarse token $\mathbf{e}_j$ takes the $(y, x, \tau)$ coordinates of the top-left latent of its patch $\mathcal{P}_j$ on the original latent grid.

### B.4 R$^2$-RoPE

In practice, we reserve eight slots for reference banks.

## C Benchmark Evaluation Protocol

### C.1 Category-Specific Evaluation

OmniXRef-Bench evaluates single-video-reference editing across Camera, Motion, Content, Style, and VFX. Video-reference editing raises three separate questions: whether the output adopts the requested dynamic attribute from the reference, whether the result looks and moves convincingly, and whether everything else from the source remains intact. A model could produce an attractive video while ignoring the requested edit, or copy the desired attribute along with unrelated reference content; neither is a successful edit. We use Gemini 3.8 Flash (Google, 2026) as a VLM judge for three separate criteria: Reference Adherence (RA) assesses the requested reference attribute, Visual Quality (VQ) assesses the output quality, and Source Preservation (SP) assesses source information that should remain unchanged, including unwanted reference-content transfer. The judge returns structured labels that we convert into scores deterministically; the complete category-specific prompts are listed in Appendix G.

### C.2 Deterministic Scoring

The judge outputs only closed-set labels, which deterministic rules convert into scores on a 0–5 scale (Table 6). Each criterion first receives a base score by averaging the scores of its labels; category-specific caps then penalize failures that the average alone would miss. Leakage is graded by the number of source attributes that the SP judge labels as imported from the reference: two or more clear imports are *major*, one clear import is *minor*, and only possible imports are *suspected*. The composite Score is

$$\mathrm{Score} = 0.45\,\mathrm{RA} + 0.25\,\min(\mathrm{VQ}, \mathrm{RA}) + 0.30\,\mathrm{SP}, \tag{6}$$

where bounding VQ by RA prevents visually appealing outputs that ignore the requested edit from scoring well, and the leakage caps prevent high scores on other criteria from compensating for severe unwanted changes. All caps are applied per example before averaging; consequently, the reported mean RA, VQ, and SP do not in general determine the mean Score, and we report VQ before the $\min(\cdot, \mathrm{RA})$ bound so that visual quality can be distinguished from editing success.

Table 6: Conversion of judge labels into scores. The label scores of the judged items are averaged into a base score, and the listed caps are then applied.

| Criterion | Judged items | Label → score | Caps |
|---|---|---|---|
| RA (Camera) | Direction, extent, timing, framing | match / partial / mismatch → $5/2.5/0$ | None (preservation and leakage are judged by SP) |
| RA (Content, Style, VFX) | Category-specific properties | match / partial / mismatch → $5/2.5/0$ | Wrong localization → 0; spillover →$\leq 2.5$; Style broken structure →$\leq 1.25$; Content clear extra import →$\leq 2.5$ |
| RA (Motion) | Pose match at six moments | match / partial / mismatch → $5/2.5/0$ | Reference share of appearance votes $\geq 0.5 \to 0$, $\geq 0.2 \to \leq 2.5$; subject absent in more than half of the moments → 0 |
| VQ | Worst defect in appeal, plausibility, coherence | none / subtle / noticeable / obvious → $5/3.75/2/0$ | None |
| SP | Source attributes | preserved / side effect / altered / absent / from reference → $5/5/2.5/1.25/0$ | Different video → 0; partly changed →$\leq 4$; leakage: suspected $\leq 3.125$, minor $\leq 1.25$, major 0 |
| Score | RA, VQ, SP | Eq. (6) | Major leakage → 0; minor leakage or SP $\leq 1.25$ →$\leq 1.25$; suspected leakage →$\leq 3.125$ |

### C.3 Camera Trajectory Errors

For each Camera case, we sample $N = 32$ temporally aligned frames from the reference and output videos and estimate their camera poses independently with VGGT-Ω (Wang et al., 2026a). Let $E_i^v$ be the estimated world-to-camera extrinsic matrix of frame $i$ in video $v \in \{\mathrm{ref}, \mathrm{out}\}$, and let $T_i^v = (E_i^v)^{-1}$ be the corresponding camera-to-world pose. We express each trajectory relative to its own first frame and normalize its translations by the maximum displacement from the first camera:

$$A_i^v = (T_0^v)^{-1} T_i^v = \begin{bmatrix} R_i^v & \mathbf{p}_i^v \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \hat{\mathbf{p}}_i^v = \frac{\mathbf{p}_i^v}{\max_j \|\mathbf{p}_j^v\|_2 + \epsilon}. \tag{7}$$

Expressing both trajectories in their own first-frame coordinates removes the arbitrary world frame of each reconstruction; no additional rigid alignment or temporal warping is applied between them. Because both relative trajectories start at the identity, the first frame contributes no error; following CamI2V (Zheng et al., 2024), we sum the per-frame errors over the remaining $N - 1 = 31$ frames:

$$\begin{aligned} \mathrm{RotErr} &= \sum_{i=1}^{N-1} \arccos\left( \frac{\mathrm{tr}\left( (R_i^{\mathrm{ref}})^\top R_i^{\mathrm{out}} \right) - 1}{2} \right), \\ \mathrm{TransErr} &= \sum_{i=1}^{N-1} \left\| \hat{\mathbf{p}}_i^{\mathrm{ref}} - \hat{\mathbf{p}}_i^{\mathrm{out}} \right\|_2, \\ \mathrm{CamMC} &= \sum_{i=1}^{N-1} \left\| [R_i^{\mathrm{ref}} \mid \hat{\mathbf{p}}_i^{\mathrm{ref}}] - [R_i^{\mathrm{out}} \mid \hat{\mathbf{p}}_i^{\mathrm{out}}] \right\|_F, \end{aligned} \tag{8}$$

where the argument of $\arccos$ is clipped to $[-1, 1]$ and RotErr is measured in radians. We then average each error over the Camera cases of OmniXRef-Bench. Since each trajectory is normal-

ized independently and camera intrinsics are not compared, these errors capture motion direction, trajectory shape, and timing, but not absolute translation magnitude or focal-length changes such as zooms.

# D Additional Quantitative Results

## D.1 Instruction-Based Editing Benchmarks

Although OmniXRef-Model focuses on reference-guided video editing, it should retain the foundational text-based editing capability on which such editing builds. We therefore evaluate T′V2V editing on VEFX-Bench (Gao et al., 2026) and ReCo-Bench (Zhang et al., 2025), following their official evaluation suites. VEFX-Bench uses VEFX-Reward to score Instruction Following (IF), Rendering Quality (RQ), and Edit Exclusivity (EE) on a 1–4 scale, together with their official geometric aggregate (GeoAgg). ReCo-Bench reports editing accuracy, video naturalness, video quality, and Total, their arithmetic mean, separately for the Add, Remove, Replace, and Style tasks, evaluated by Gemini-2.5-Flash.

We report T′V2V results on VEFX-Bench (Gao et al., 2026) in Table 7. Overall, OmniXRef-Model outperforms the open-source baselines (Wei et al., 2025; Jiang et al., 2025), as well as Wan 2.6 (Alibaba Cloud, 2025), and performs on par with commercial models such as Grok Imagine (xAI, 2026) and Seedance 2.0 (ByteDance Seed, 2026a). This indicates that, although OmniXRef-Model focuses on reference-guided video editing, it retains the foundational text-based editing capability.

Table 7: VEFX-Bench leaderboard for T′V2V.

| Model | IF ↑ | RQ ↑ | EE ↑ | GeoAgg ↑ |
|---|---|---|---|---|
| Commercial | | | | |
| Kling o3 Omni | 3.033 | 3.588 | 3.043 | 3.057 |
| Kling o1 | 3.040 | 3.534 | 2.976 | 2.985 |
| Runway Gen-4.5 | 2.817 | 3.319 | 2.923 | 2.912 |
| Seedance 2.0 | 2.811 | 3.421 | 3.088 | 2.766 |
| Grok Imagine | 2.606 | 3.346 | 3.376 | 2.723 |
| Wan 2.6 | 2.012 | 3.317 | 2.446 | 2.146 |
| Open-source | | | | |
| UniVideo | 2.294 | 3.266 | **3.091** | 2.516 |
| VACE | 2.027 | 3.172 | 1.180 | 1.775 |
| **OmniXRef** | **2.699** | **3.315** | 2.615 | **2.708** |

Table 8 reports the ReCo-Bench leaderboard, and boldface marks the best value within each task and metric.

## D.2 Camera Trajectory Evaluation

To complement the implicit VLM-based evaluation, we explicitly evaluate camera following with trajectory errors computed from camera poses estimated by VGGT-Ω (Wang et al., 2026a), as defined in Appendix C.3. As shown in Table 9, OmniXRef-Model substantially outperforms both open-source and commercial baselines on all three metrics, demonstrating its strong reference adherence.

## D.3 Per-Category Results on OmniXRef-Bench

Tables 10 and 11 expand Tables 2 and 3, respectively, by reporting Reference Adherence (RA), Visual Quality (VQ), Source Preservation (SP), and the composite Score for each category of OmniXRef-Bench. They use the same evaluation cases, category weighting, and boldface convention as the main-text tables, and dashes denote unavailable results.

Table 8: ReCo-Bench leaderboard for T′V2V across four tasks: Add, Remove, Replace, Style.

| Model | Editing accuracy ↑ | Video naturalness ↑ | Video quality ↑ | Total ↑ |
|---|---|---|---|---|
| *Add* | | | | |
| InsViE | 2.60 | 3.10 | 3.46 | 3.05 |
| Lucy-Edit | 6.47 | 5.70 | 6.77 | 6.31 |
| Ditto | 6.70 | 7.57 | 8.41 | 7.56 |
| ReCo | 8.54 | 7.55 | 8.61 | 8.23 |
| ReCo_Ref | 7.96 | 7.39 | 8.69 | 8.02 |
| VInO | 9.44 | 8.00 | 9.11 | 8.85 |
| Mamoda2.5 | 9.31 | 8.27 | 8.83 | 8.80 |
| **OmniXRef** | **9.55** | **8.56** | **9.12** | **9.08** |
| *Remove* | | | | |
| InsViE | 2.44 | 3.76 | 3.29 | 3.16 |
| VACE | 4.57 | 5.43 | 5.56 | 5.19 |
| ReCo | 7.28 | 6.90 | 6.82 | 7.00 |
| ReCo_Ref | 8.06 | 7.45 | 7.44 | 7.65 |
| VInO | 8.39 | 8.00 | 8.10 | 8.16 |
| Mamoda2.5 | **9.15** | **8.50** | **8.35** | **8.67** |
| **OmniXRef** | 7.77 | 7.46 | 7.30 | 7.51 |
| *Replace* | | | | |
| InsViE | 2.10 | 3.91 | 3.49 | 3.17 |
| Lucy-Edit | 7.08 | 6.21 | 6.88 | 6.72 |
| Ditto | 4.56 | 7.21 | 7.96 | 6.58 |
| ReCo | 9.43 | 8.01 | 8.77 | 8.74 |
| ReCo_Ref | 8.35 | 7.67 | 8.38 | 8.13 |
| VInO | 9.17 | 8.18 | 8.74 | 8.70 |
| Mamoda2.5 | **9.60** | 8.64 | 8.95 | **9.06** |
| **OmniXRef** | 9.10 | **8.70** | **8.96** | 8.92 |
| *Style* | | | | |
| InsViE | 8.17 | 8.21 | 7.35 | 7.91 |
| Lucy-Edit | 4.65 | 4.67 | 5.17 | 4.83 |
| Ditto | 9.20 | 9.07 | 8.77 | 9.01 |
| ReCo | **9.42** | 9.19 | 8.90 | **9.17** |
| ReCo_Ref | 8.81 | **9.27** | **9.29** | 9.12 |
| VInO | 9.05 | 9.25 | 9.01 | 9.10 |
| Mamoda2.5 | 8.95 | 9.19 | 8.83 | 8.99 |
| **OmniXRef** | 8.37 | 8.80 | 8.39 | 8.52 |

Table 9: Camera trajectory errors on OmniXRef-Bench, summed over frames; RotErr is in radians.

| Model | RotErr↓ | TransErr↓ | CamMC↓ |
|---|---|---|---|
| CamCloneMaster | 2.086 | 9.329 | 10.332 |
| **OmniXRef** | **0.874** | **3.961** | **4.437** |
| COMMERCIAL | | | |
| Seedance 2.0 | 4.443 | 22.455 | 24.008 |
| Seedance 2.5 | 4.373 | 22.445 | 24.001 |

Table 10: Per-category VLM evaluation of video-reference attribute transfer on OmniXRef-Bench, expanding Table 2 with Reference Adherence (RA), Visual Quality (VQ), and Source Preservation (SP) for each category.

| Model | Content | | | | Style | | | | VFX | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RA | VQ | SP | Score | RA | VQ | SP | Score | RA | VQ | SP | Score |
| OPEN-SOURCE | | | | | | | | | | | | |
| RefVFX | 0.94 | 3.35 | 4.55 | 1.98 | 1.12 | 3.55 | **4.90** | 2.24 | 1.69 | **3.40** | 4.39 | 2.49 |
| OURS | | | | | | | | | | | | |
| **OmniXRef** | **4.69** | **3.87** | **4.70** | **4.48** | **3.71** | **3.60** | 4.65 | **3.74** | **3.56** | 3.25 | **4.52** | **3.55** |
| COMMERCIAL | | | | | | | | | | | | |
| MiniMax-H3 | 3.81 | 4.03 | 1.97 | 2.23 | 2.92 | 3.92 | 1.75 | 0.94 | 3.09 | 3.56 | 1.90 | 1.32 |
| Seedance 2.0 | 4.69 | 3.93 | 4.03 | 3.73 | 3.71 | 3.85 | 3.95 | 3.22 | 3.69 | 3.72 | 3.74 | 2.92 |
| Seedance 2.5 | 4.81 | 3.76 | 4.25 | 4.04 | 3.52 | 3.47 | 3.02 | 2.30 | 3.78 | 3.50 | 2.84 | 2.22 |

| Model | Camera | | | | Motion | | | | Overall avg. | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RA | VQ | SP | Score | RA | VQ | SP | Score | RA | VQ | SP | Score |
| OPEN-SOURCE | | | | | | | | | | | | |
| CamCloneMaster | 2.52 | **3.74** | 4.99 | 3.20 | – | – | – | – | – | – | – | – |
| MotionFollower | – | – | – | – | 2.02 | 2.43 | 2.99 | 2.14 | – | – | – | – |
| OURS | | | | | | | | | | | | |
| **OmniXRef** | **3.48** | **3.74** | **5.00** | **3.81** | **2.86** | **3.88** | **4.57** | **3.27** | **3.66** | **3.67** | **4.69** | **3.77** |
| COMMERCIAL | | | | | | | | | | | | |
| MiniMax-H3 | 2.44 | 3.89 | 3.67 | 2.07 | 0.05 | 3.82 | 2.23 | 0.69 | 2.46 | 3.84 | 2.30 | 1.45 |
| Seedance 2.0 | 2.65 | 3.81 | 4.97 | 3.27 | 1.85 | 3.84 | 4.18 | 2.45 | 3.32 | 3.83 | 4.17 | 3.12 |
| Seedance 2.5 | 2.33 | 3.89 | 4.98 | 3.07 | 1.58 | 3.86 | 3.45 | 2.00 | 3.21 | 3.70 | 3.71 | 2.73 |

Table 11: Per-category ablations of OmniXRef-Model on OmniXRef-Bench, expanding Table 3 with Reference Adherence (RA), Visual Quality (VQ), and Source Preservation (SP) for each category.

| Variant | Content | | | | Style | | | | VFX | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RA | VQ | SP | Score | RA | VQ | SP | Score | RA | VQ | SP | Score |
| w/o Mix-Patchify | 3.75 | 3.80 | 3.30 | 2.97 | 3.40 | 3.22 | 3.58 | 2.71 | **4.12** | **3.61** | 3.64 | 3.30 |
| Reference-bank dropout | 3.06 | 3.73 | 3.30 | 2.48 | 3.25 | 3.50 | 4.45 | 3.29 | 3.72 | 3.47 | 4.14 | 3.42 |
| w/o $R^2$-RoPE | 3.31 | 3.47 | 3.89 | 3.06 | 2.83 | **3.62** | 4.44 | 3.14 | 3.41 | 3.48 | **4.66** | **3.64** |
| w/o curriculum learning | 3.69 | 3.56 | 3.70 | 3.07 | 3.08 | 3.44 | 4.32 | 3.15 | 3.69 | 3.37 | 4.22 | 3.51 |
| **OmniXRef (full)** | **4.69** | **3.87** | **4.70** | **4.48** | **3.71** | 3.60 | **4.65** | **3.74** | 3.56 | 3.25 | 4.52 | 3.55 |

| Variant | Camera | | | | Motion | | | | Overall avg. | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RA | VQ | SP | Score | RA | VQ | SP | Score | RA | VQ | SP | Score |
| w/o Mix-Patchify | 3.01 | **3.87** | 4.67 | 3.24 | 1.51 | 3.86 | 3.92 | 2.11 | 3.16 | **3.67** | 3.82 | 2.86 |
| Reference-bank dropout | 2.95 | 3.40 | 4.25 | 2.85 | 2.07 | 3.80 | 3.43 | 2.11 | 3.01 | 3.58 | 3.91 | 2.83 |
| w/o $R^2$-RoPE | 3.20 | 3.69 | 4.94 | 3.61 | 1.57 | 3.74 | **4.60** | 2.44 | 2.86 | 3.60 | 4.50 | 3.18 |
| w/o curriculum learning | 2.69 | 3.67 | 4.97 | 3.32 | 1.81 | **3.92** | 4.53 | 2.57 | 2.99 | 3.59 | 4.35 | 3.13 |
| **OmniXRef (full)** | **3.48** | 3.74 | **5.00** | **3.81** | **2.86** | 3.88 | 4.57 | **3.27** | **3.66** | **3.67** | **4.69** | **3.77** |

## E Inference Latency and Computational Cost

We measure the DiT inference performance on a single NVIDIA A100 GPU, with the source and reference videos at 480P and 121 frames. We evaluate inference latency and computational cost along two dimensions: the number of references and the reference patch size in Mix-Patchify (Section 3.2). As shown in Figure 7, both latency and computational cost grow with the number of DiT tokens: more references and finer patches produce longer sequences and thus higher cost. Mix-Patchify substantially reduces both, and its advantage widens as the number of references increases.

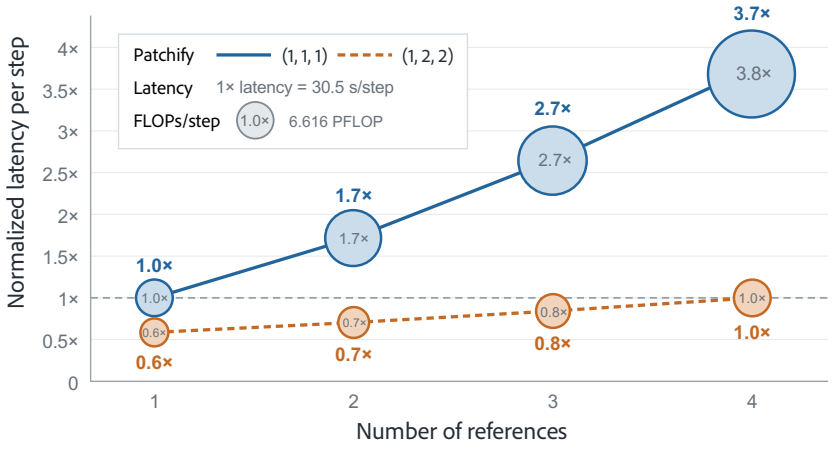



Figure 7: Analysis of the inference latency and computational cost.

## F Limitations

OmniXRef-Model supports attribute-level control but cannot yet reliably disentangle finer-grained factors within an attribute. For example, the model may transfer facial expressions and body poses together from a motion reference, making it difficult to apply either independently (Figure 8). Scaling to larger reference compositions is also constrained by the difficulty of constructing high-quality training data in which the target video incorporates all requested edits. Although our reserved-slot design supports generalization beyond the reference counts seen during training, performance degrades substantially with four or more references. Our training data also underrepresent content beyond natural scenes, particularly VFX involving text rendering, indicating a need for greater diversity in visual content and editing operations. Finally, training is currently limited to resolutions up to 480P, and the effectiveness of our framework for higher-resolution editing at 720P or 4K remains unexplored.




Figure 8: Coupled transfer of facial expression and body pose from a Motion reference.

## G VLM Judge Prompts

This section lists the complete prompts that Gemini 3.8 Flash (Google, 2026) receives when scoring OmniXRef-Bench. Each category has its own Reference Adherence (RA) and Source Preservation (SP) prompts, which state explicitly what may change, what must be preserved, and which side

effects of the edit are permitted; the Visual Quality (VQ) prompt is shared across categories. Placeholders in braces, such as `{instruction}` and `{transferred_attribute}`, are filled from the annotation of each case, and `[attach: ...]` marks where the visual evidence is inserted. The judge never outputs numeric scores: it returns only closed-set labels and concrete, locatable evidence, from which the scores are computed deterministically as described in Appendix C. The RA, VQ, and SP calls for a case are independent and share no context. Visual evidence is provided as time-aligned frame grids.

### G.1 CAMERA

The RA prompt compares only the camera motion of the reference and the output, using a two-row grid with eight time-aligned reference and output frames. The equal-weight average of its four property labels gives the Camera RA. Preservation of the source content and leakage of non-camera content from the reference are judged solely by the SP prompt.

**Camera – Reference Adherence (RA)**

**System**

You judge whether output O follows the camera trajectory of reference R. R and O show different scenes by design. Compare camera movement only, not subjects, scene content, colour, or absolute object positions. Use background movement and parallax where visible; moving subjects alone are not evidence of camera movement.

Return one JSON object and nothing else. Do not output a numeric score.

**User**

*instruction* "{instruction}"

[attach: RO_rows — one image, 2 rows x 8 columns.
Top = R (reference), bottom = O (output).
Read each row from left to right.]

**Step 1.** Describe R's camera path and O's camera path concretely: direction, amount of viewpoint change, when the move starts/turns/stops, and how the view changes. → "R_reading", "O_reading"

**Step 2.** Compare O's camera movement against R's on four properties:

*direction* dominant direction and movement type (e.g. pan, tilt, dolly, orbit, zoom); do not confuse subject motion with camera motion
*extent* comparable relative amount of viewpoint change; do not demand equal pixel displacement in different scenes
*timing* comparable onset, pauses, turns and speed progression over normalized time; do not penalize different clip durations
*framing* comparable change in viewpoint or shot scale from start to end; do not compare absolute scene composition

For each property use "match", "partial", or "mismatch". Judge only what the frames support; do not invent an exact 3D trajectory.

**Return only:**

```
{"R_reading":"","O_reading":"",
 "per_property":{"direction":"match|partial|mismatch",
                 "extent":"match|partial|mismatch",
                 "timing":"match|partial|mismatch",
                 "framing":"match|partial|mismatch"},
 "reasoning":""}
```

**Camera – Source Preservation (SP)**

**System**

You judge preservation for camera-trajectory transfer. O should retain S's scene and content while its camera trajectory changes. Check whether S's non-camera content survives and whether non-camera content from R was copied into O.

Changes in viewpoint, crop, perspective, parallax, scale and visibility caused by the new camera are allowed. The requested camera movement from R is not reference-content leakage.

Return one JSON object and nothing else. Do not output a numeric score.

**User**

*instruction* "{instruction}"

[attach: SOR_rows — one image, 3 rows x 6 columns.
Top = S (source), middle = O (output), bottom = R (reference).
Columns correspond to the same normalized moments.
Compare O with S for preservation; use R to identify copied content.]

**Part A.** Ignoring the allowed camera change, does O still depict S's video? Choose "preserved", "partly_changed", or "different_video". Give one concrete reason. → "preservation_overall", "overall_reason"

**Part B.** List the checkable attributes of S using these names:

*setting* place and environment
*subjects* people/animals and their identity and appearance
*objects* notable props and objects
*lighting* lighting direction and character
*style* palette and visual treatment
*action* what the subjects do

For each, describe its version in S and O, then choose:

*"preserved"* O keeps S's version
*"side_effect"* changed only because of the new viewpoint, crop or occlusion
*"altered"* changed into something else
*"from_reference"* distinctive non-camera content from R appeared in O
*"absent"* gone despite still being checkable from O's new viewpoint

Use "clear" or "possible" for certainty. Skip an attribute if the new viewpoint makes it impossible to check. Do not list camera or framing: they are the intended edit.

**Return only:**

```
{"preservation_overall":"preserved|partly_changed|different_video",
 "overall_reason":"",
 "attributes":[{"name":"setting|subjects|objects|lighting|style|action",
                "in_source":"","in_output":"",
                "verdict":"preserved|side_effect|altered|from_reference|absent",
                "certainty":"clear|possible"}]}
```

### G.2 MOTION

For each of six time-aligned moments, the RA judge receives one image with the source, reference, and output frames side by side (S | R | O) in a separate call. The six pose labels are averaged, and the 18 appearance votes (three attributes over six moments) determine whether the subject's identity was copied from the reference.

**Motion – Reference Adherence (RA), one call per sampled moment**

**System**

You are shown one moment from three videos, side by side at the same normalized time:

*S* source (the person whose look must be kept)
*R* reference (the person whose motion must be copied)
*O* output (the edited result)

S and R show different people and different places by design.

Judge this single moment only. Two things are being judged and they must never be confused:

*Pose* the body configuration at this instant. O should match R.
*Identity* what the subject looks like. O should match S, not R.

A very close pose match is a success, never evidence that identity was copied.

Return one JSON object and nothing else. Do not output any numeric score.

**User**

*instruction* "{instruction}"
*motion* "{transferred_attribute}"

[attach: one image, 1 row x 3 columns – S | R | O at the same moment]

**Step 1.** Describe R's body configuration at this instant: torso orientation, where each limb is, head direction, and what the action looks like frozen here. → "R_pose"

**Step 2.** Describe O's body configuration the same way. → "O_pose"

**Step 3.** "pose_match": how well does O's pose reproduce R's pose at this moment?

*"match"* the same configuration; limbs in the same places
*"partial"* the same general action but limbs or orientation clearly differ
*"mismatch"* a different configuration, or O's subject is doing something else

Judge the body only. A different person, different clothes or a different room must not lower this label.

**Step 4.** Now ignore all pose. Looking only at how the subject looks, say for each attribute whether O's subject resembles S's or R's:

*clothing* garment type and colour
*hair* colour, length, style
*build* body type and proportions

Use "unclear" only when the subject is too small or occluded to tell.

**Step 5.** "subject_present": is there a recognisable subject performing an action in O at all?

*"yes"* | "no"

**Return only:**

```
{"R_pose":"","O_pose":"",
 "pose_match":"match|partial|mismatch",
 "appearance":{"clothing":"source|reference|unclear","hair":"source|reference|unclear",
               "build":"source|reference|unclear"},
 "subject_present":"yes|no",
 "reasoning":""}
```

## Motion – Source Preservation (SP)

**System**

You judge preservation for motion transfer: whether an edited video kept everything it was supposed to keep.

The requested edit is:

*instruction* "{instruction}"
*transferred attribute* "{transferred_attribute}"

O should be S's video with only the transferred attribute changed. Two kinds of failure matter:

*(a)* something of S's was lost or altered when it should have stayed;
*(b)* something of R's, other than the transferred attribute, turned up in O.

The transferred attribute appearing in O is correct, however strongly it resembles R's version. The subject's pose and limb positions will change — that is the requested edit and is never a preservation failure. The subject's identity and appearance changing is a failure, and is the most important thing you are checking.

Judge from the pixels. Return one JSON object and nothing else. Do not output any numeric score.

**User**

[attach: SOR_rows] (one image, 3 rows x 6 columns, columns are time-aligned:
Top row = S, the source, at 6 timestamps, earliest on the left
Middle row = O, the output, at the same 6 timestamps
Bottom row = R, the reference, at the same 6 timestamps
Read along a row to see how that video evolves over time.
Compare O with S by looking one row up, and O with R one row down.)

**Part A.** one overall judgement for the whole clip. Compare the middle row against the top row. Mentally set aside everything the edit was allowed to change. Ignoring that completely, does what remains still look like the same video?

*"preservation_overall"* "preserved" | "partly_changed" | "different_video"
*"overall_reason"* one sentence saying what drove that answer.

**Part B.** attribute-by-attribute comparison.

**Step B1.** Read the top row (S) and list the key attributes S actually has. Use these slots, skipping any that do not apply to this clip:

*identity* the source subject's face, hair, clothing and body type
*setting* where it takes place: background, terrain, architecture, interior
*objects* notable props and objects
*layout* spatial arrangement, composition, framing
*camera* camera position and how it moves along the row
*lighting* lighting direction, tone, time of day

Describe S's version of each in a few concrete words → "in_source"

**Step B2.** For that same attribute, compare the middle row (O) against the top row (S) and against the bottom row (R), then choose one verdict:

*"preserved"* O keeps S's version of it
*"side_effect"* changed only as an unavoidable consequence of the transferred attribute (a change forced by the new pose, such as a limb now occluding something)
*"altered"* changed into something that is neither S's nor R's
*"from_reference"* O's version of it now matches R's instead of S's
*"absent"* it is gone from O

State your certainty — "clear" | "possible" — and describe what you actually see in O in a few concrete words → "in_output"
Do not list the motion, pose or limb positions — they are the transferred attribute. Judge 'identity' strictly: if O's subject wears R's clothes, has R's hair or R's build, that is 'from_reference', not 'altered'.

**Return only:**

```
{"preservation_overall":"preserved|partly_changed|different_video",
 "overall_reason":"",
 "attributes":[{"name":"identity|setting|objects|layout|camera|lighting",
                "in_source":"","in_output":"",
                "verdict":"preserved|side_effect|altered|from_reference|absent",
                "certainty":"clear|possible"}]}
```

### G.3 CONTENT

The RA prompt receives six-frame filmstrips of the reference and the output, together with the location of the requested element in each.

**Content – Reference Adherence (RA)**

**System**

You compare a reference clip R and an output clip O to decide whether one specific content element from R was correctly brought into O. That element may be a background, an object, a subject, or a visual attribute — the instruction says which. Compare only that element; never compare the rest of R's scene.

Bringing it in is the point of the edit. However strongly O resembles R on that one element, that is success, never evidence of copying.

Return one JSON object and nothing else. Do not output any numeric score — scores are computed downstream from your labels.

**User**

*instruction* "{instruction}"
*element* "{attribute}" (the content element that was requested)
*locus_R* "{ref_locus}" (in R, where the element is)
*locus_O* "{edit_locus}" (in O, where it should appear)

[attach: R_filmstrip] (6 frames)

[attach: O_filmstrip] (6 frames)

**Step 1.** Read the instruction and choose exactly one edit type:

*"replace_background"* R's surroundings take over S's surroundings
*"replace_object"* one thing in the source is swapped for a thing from R
*"add_object"* a thing from R is added; nothing existing is removed
*"apply_attribute"* a property from R (pattern, colour, material) is applied to something that stays

→ "edit_type"

**Step 2.** From R at locus_R, describe the element: what it is, its texture, colour, material and distinguishing details. → "R_reading"

**Step 3.** From O, state whether the element is present and describe it the same way. → "O_reading"

**Step 4.** For each property decide whether O matches R:

*identity* it is recognisably the same thing
*appearance* texture, colour, material detail match
*extent* does it occupy the right scope in O? Judge by edit_type: replace_background → it fills the surroundings, and O's foreground subjects still sit in front of it, not erased
  *replace_object* → right position, scale and orientation, standing where the thing it replaced used to be
  *add_object* → right position, scale and orientation for O's scene
  *apply_attribute* → applied to the thing named in the instruction, and to nothing else
*integration* lighting, shadow, contact and occlusion are consistent with O's scene

A different underlying scene in R is expected and must never lower these labels.

**Step 5.** Set "extra_import": besides the requested element, did anything else from R appear in O (R's other objects, R's subjects, R's surroundings, R's camera behaviour)? → "none" | "possible" | "clear"

**Step 6.** Set "localization":

*"correct"* only the intended region changed
*"leaks"* the change spilled into regions it should not touch
*"wrong"* the element is absent, or landed somewhere else entirely

**Return only:**

```
{"edit_type":"replace_background|replace_object|add_object|apply_attribute",
 "R_reading":"","O_reading":"",
 "per_property":{"identity":"match|partial|mismatch","appearance":"match|partial|mismatch",
                 "extent":"match|partial|mismatch","integration":"match|partial|mismatch"},
 "localization":"correct|leaks|wrong",
 "extra_import":"none|possible|clear",
 "reasoning":""}
```

**Content – Source Preservation (SP)**

**System**

You judge preservation for content transfer: whether an edited video kept everything it was supposed to keep.

The requested edit is:

*instruction* "{instruction}"
*transferred attribute* "{transferred_attribute}"

O should be S's video with only the transferred attribute changed. Two kinds of failure matter:

*(a)* something of S's was lost or altered when it should have stayed;
*(b)* something of R's, other than the transferred attribute, turned up in O.

The transferred attribute appearing in O is correct, however strongly it resembles R's version. This task deliberately imports one element from R: the requested element above. That element arriving from R is correct and must never be reported as 'from_reference'. Anything else arriving from R is a failure.

Judge from the pixels. Return one JSON object and nothing else. Do not output any numeric score.

**User**

[attach: SOR_rows] (one image, 3 rows x 6 columns, columns are time-aligned:
Top row = S, the source, at 6 timestamps, earliest on the left
Middle row = O, the output, at the same 6 timestamps
Bottom row = R, the reference, at the same 6 timestamps
Read along a row to see how that video evolves over time.
Compare O with S by looking one row up, and O with R one row down.)

**Part A.** one overall judgement for the whole clip. Compare the middle row against the top row. Mentally set aside everything the edit was allowed to change. Ignoring that completely, does what remains still look like the same video?

*"preservation_overall"* "preserved" | "partly_changed" | "different_video"
*"overall_reason"* one sentence saying what drove that answer.

**Part B.** attribute-by-attribute comparison.

**Step B1.** Read the top row (S) and list the key attributes S actually has. Use these slots, skipping any that do not apply to this clip:

*setting* where it takes place: background, terrain, architecture, interior
*subjects* the people / animals / main entities present, and their identity
*objects* notable props and objects other than the requested element
*layout* spatial arrangement, composition, framing
*camera* camera position and how it moves along the row
*lighting* lighting direction, tone, time of day
*action* what happens across the six columns

Describe S's version of each in a few concrete words → "in_source"

**Step B2.** For that same attribute, compare the middle row (O) against the top row (S) and against the bottom row (R), then choose one verdict:

*"preserved"* O keeps S's version of it
*"side_effect"* changed only as an unavoidable consequence of the transferred attribute (the requested element occluding or casting light on something)
*"altered"* changed into something that is neither S's nor R's
*"from_reference"* O's version of it now matches R's instead of S's
*"absent"* it is gone from O

State your certainty — "clear" | "possible" — and describe what you actually see in O in a few concrete words → "in_output"

Do not list the requested element itself — its arrival from R is the point of the edit.

**Return only:**

```
{"preservation_overall":"preserved|partly_changed|different_video",
 "overall_reason":"",
 "attributes":[{"name":"setting|subjects|objects|layout|camera|lighting|action",
                "in_source":"","in_output":"",
                "verdict":"preserved|side_effect|altered|from_reference|absent",
                "certainty":"clear|possible"}]}
```

### G.4 STYLE

The RA prompt receives six rows of time-aligned reference and output frame pairs, and additionally checks whether the stylization destroys the structure of the output.

**Style – Reference Adherence (RA)**

**System**

You compare two clips, R (reference) and O (output). R carries a specific lighting/colour style, and O was supposed to receive that same style. Decide how faithfully O reproduces it. The two clips show different scenes by design — compare only the style itself, never the surrounding content, and never penalise O for having a different scene than R. Return one JSON object and nothing else. Do not output any numeric score.

**User**

*instruction* "{instruction}"
*style* "{transferred_attribute}"

scope: whole frame

[attach: RO_pairs] (6 rows; each row is R[t] on the left, O[t] on the right,
At the same normalized time)

**Step 1.** From R, describe the style concretely on each property — palette, lighting, texture. → "R_reading"

**Step 2.** In O, describe the same three properties. → "O_reading"

**Step 3.** For each property decide whether O matches R. Judge the style only; a different underlying scene is expected and must not lower these labels.

**Step 4.** "structure_check": judging O alone, are its objects and surfaces structurally coherent ("intact"), mildly deformed or smeared ("soft"), or clearly distorted, melted or unrecognisable ("broken")? Over-stylisation that destroys structure must be caught

*here* a beautiful grade over ruined geometry is not a success.

**Return only:**

```
{"R_reading":"","O_reading":"",
 "per_property":{"palette":"match|partial|mismatch","lighting":"match|partial|mismatch",
                 "texture":"match|partial|mismatch"},
 "structure_check":"intact|soft|broken",
 "reasoning":""}
```

**Style – Source Preservation (SP)**

**System**

You judge preservation for style transfer: whether an edited video kept everything it was supposed to keep.

The requested edit is:

*instruction* "{instruction}"
*transferred attribute* "{transferred_attribute}"

O should be S's video with only the transferred attribute changed. Two kinds of failure matter:

*(a)* something of S's was lost or altered when it should have stayed;
*(b)* something of R's, other than the transferred attribute, turned up in O.

The transferred attribute appearing in O is correct, however strongly it resembles R's version. Palette, colour grading, contrast, lighting tone and surface texture are the transferred attribute — they are supposed to change, and changing them is never a failure. What must survive is structure: the objects, their shapes and positions, and the place itself.

Judge from the pixels. Return one JSON object and nothing else. Do not output any numeric score.

**User**

[attach: SOR_rows] (one image, 3 rows x 6 columns, columns are time-aligned:
Top row = S, the source, at 6 timestamps, earliest on the left
Middle row = O, the output, at the same 6 timestamps
Bottom row = R, the reference, at the same 6 timestamps
Read along a row to see how that video evolves over time.
Compare O with S by looking one row up, and O with R one row down.)

**Part A.** one overall judgement for the whole clip. Compare the middle row against the top row. Mentally set aside everything the edit was allowed to change. Ignoring that completely, does what remains still look like the same video?

*"preservation_overall"* "preserved" | "partly_changed" | "different_video"
*"overall_reason"* one sentence saying what drove that answer.

**Part B.** attribute-by-attribute comparison.

**Step B1.** Read the top row (S) and list the key attributes S actually has. Use these slots, skipping any that do not apply to this clip:

*setting* where it takes place: background, terrain, architecture, interior
*subjects* the people / animals / main entities present, and their identity
*objects* notable props and objects
*layout* spatial arrangement, composition, framing
*camera* camera position and how it moves along the row
*action* what happens across the six columns

Describe S's version of each in a few concrete words → "in_source"

**Step B2.** For that same attribute, compare the middle row (O) against the top row (S) and against the bottom row (R), then choose one verdict:

*"preserved"* O keeps S's version of it
*"side_effect"* changed only as an unavoidable consequence of the transferred attribute (a colour or tonal shift that the new style necessarily implies)
*"altered"* changed into something that is neither S's nor R's
*"from_reference"* O's version of it now matches R's instead of S's
*"absent"* it is gone from O

State your certainty — "clear" | "possible" — and describe what you actually see in O in a few concrete words → "in_output"

Do not list lighting, colour, palette or texture — they are the transferred attribute. A useful test: ignore all colour and lighting. Does the remaining structure still look like S's?

**Return only:**

```
{"preservation_overall":"preserved|partly_changed|different_video",
 "overall_reason":"",
 "attributes":[{"name":"setting|subjects|objects|layout|camera|action",
                "in_source":"","in_output":"",
                "verdict":"preserved|side_effect|altered|from_reference|absent",
                "certainty":"clear|possible"}]}
```

### G.5 VFX

The RA prompt receives a two-row grid with six time-aligned reference and output frames, and additionally labels where the effect appears and how long it remains visible.

**VFX – Reference Adherence (RA)**

**System**

You compare two clips, R (reference) and O (output). R exhibits a specific visual effect, and O was supposed to receive that same effect. Decide how faithfully O reproduces it. The two clips show different scenes by design — compare only the effect itself, never the surrounding content, and never penalise O for having a different scene than R. Return one JSON object and nothing else. Every field is a label chosen from the listed options or free text — never a number, never a rating.

**User**

*instruction* "{instruction}"
*effect* "{transferred_attribute}"

target_region:"{edit_locus}" (the region of O where the effect belongs)

[attach: RO_strips] (one image, 2 rows x 6 columns.
Top row = R at 6 timestamps, earliest on the left
Bottom row = O at the same 6 timestamps
Read each row left-to-right to see how the effect evolves;
Compare top and bottom within a column for the same moment.)

**Step 1.** Reading the top row left to right, describe R's effect concretely on each property — appearance, colour, dynamics, integration. → "R_reading"

**Step 2.** Reading the bottom row the same way, state whether the effect is present in O and, if so, describe it on the same four properties. → "O_reading"

**Step 3.** For each property decide whether O matches R. Judge the effect only; a different underlying scene is expected and must not lower these labels.

*appearance* the form, shape and material of the effect
*color* its colour and colour temperature
*dynamics* how it evolves across the row: onset, spread, decay, speed
*integration* how it sits in the scene: lighting it casts, shadows, occlusion

**Step 4.** "localization": is the effect in O in the target_region ("correct"), somewhere else ("wrong"), or does it also bleed into regions it should not touch ("leaks")? If the effect is entirely absent, use "wrong".

**Step 5.** "effect_duration": across the 6 bottom-row frames, how long is the effect clearly visible?

*"absent"* not visible in any frame
*"brief"* visible in only one or two frames
*"intermittent"* comes and goes across the row
*"sustained"* clearly visible in most or all frames

A correct effect may legitimately be brief; this label is not a quality rating.

**Return only:**

```
{"R_reading":"","O_reading":"",
 "per_property":{"appearance":"match|partial|mismatch","color":"match|partial|mismatch",
                 "dynamics":"match|partial|mismatch","integration":"match|partial|mismatch
    "},
 "localization":"correct|wrong|leaks",
 "effect_duration":"absent|brief|intermittent|sustained",
 "reasoning":""}
```

## VFX – Source Preservation (SP)

**System**

You judge preservation for local reference-based effect transfer: whether an edited video kept everything it was supposed to keep.

The requested edit is:

*instruction* "{instruction}"
*transferred attribute* "{transferred_attribute}"

O should be S's video with only the transferred attribute changed. Two kinds of failure matter:

*(a)* something of S's was lost or altered when it should have stayed;
*(b)* something of R's, other than the transferred attribute, turned up in O.

The transferred attribute appearing in O is correct, however strongly it resembles R's version. Changes that are a physical consequence of it — light it casts, reflections, shadows, what it occludes — are permitted and are not failures.

Judge from the pixels. Return one JSON object and nothing else. Do not output any numeric score.

**User**

[attach: SOR_rows] (one image, 3 rows x 6 columns, columns are time-aligned:
Top row = S, the source, at 6 timestamps, earliest on the left
Middle row = O, the output, at the same 6 timestamps
Bottom row = R, the reference, at the same 6 timestamps
Read along a row to see how that video evolves over time.
Compare O with S by looking one row up, and O with R one row down.)

**Part A.** one overall judgement for the whole clip. Compare the middle row against the top row. Mentally set aside everything the edit was allowed to change. Ignoring that completely, does what remains still look like the same video?

*"preservation_overall"* "preserved" | "partly_changed" | "different_video"
*"overall_reason"* one sentence saying what drove that answer.

**Part B.** attribute-by-attribute comparison.

**Step B1.** Read the top row (S) and list the key attributes S actually has. Use these slots, skipping any that do not apply to this clip:

*setting* where it takes place: background, terrain, architecture, interior
*subjects* the people / animals / main entities present, and their identity
*objects* notable props and objects
*layout* spatial arrangement, composition, framing
*camera* camera position and how it moves along the row
*lighting* lighting direction, tone, time of day
*action* what happens across the six columns

Describe S's version of each in a few concrete words → "in_source"

**Step B2.** For that same attribute, compare the middle row (O) against the top row (S) and against the bottom row (R), then choose one verdict:

*"preserved"* O keeps S's version of it
*"side_effect"* changed only as an unavoidable consequence of the transferred attribute (light spill, reflection, shadow, occlusion)
*"altered"* changed into something that is neither S's nor R's
*"from_reference"* O's version of it now matches R's instead of S's

*"absent"* it is gone from O

State your certainty — "clear" | "possible" — and describe what you actually see in O in a few concrete words → "in_output"
Do not list the transferred attribute itself as an attribute.

**Return only:**

```
{"preservation_overall":"preserved|partly_changed|different_video",
 "overall_reason":"",
 "attributes":[{"name":"setting|subjects|objects|layout|camera|lighting|action",
                "in_source":"","in_output":"",
                "verdict":"preserved|side_effect|altered|from_reference|absent",
                "certainty":"clear|possible"}]}
```

### G.6 VISUAL QUALITY

The VQ prompt is shared by all categories and receives only six frames of the output; withholding the source and reference prevents the judge from conflating visual quality with reference similarity.

**All Categories – Visual Quality (VQ)**

**System**

You assess the intrinsic quality of one video clip. Your job is to enumerate concrete, locatable defects — things you can actually point at in the frames. Do not give an overall impression, and do not rate the clip. Use the exact words offered below for "property" and "obviousness"; do not substitute synonyms. Return one JSON object and nothing else. Do not output any numeric score.

**User**

[attach: clip_strip] (6 frames of the clip, earliest on the left)

**Step 1.** Describe what the clip shows, in one or two sentences. → "observation"

**Step 2.** Hunt for defects. Emit one entry per defect you can actually point at. Emit as many entries as you find, or none at all → "defects"

*"property"* exactly one of the three, the one it damages most:
- *appeal* aesthetic craft: composition, lighting, color, framing, overall production value
- *plausibility* does it look real: warping, melting, extra or missing limbs/fingers, broken text, flicker, ghosting, smeared texture, impossible anatomy or physics
- *coherence* does it hold together across the 6 frames: identity drift, objects appearing/vanishing, lighting or color jumps, scale or perspective inconsistency

*"what"* the defect and where it is, concretely (e.g. "the left hand blurs into three fingers in frames 3-4")

*"obviousness"* exactly one of:
- *"subtle"* you had to look for it
- *"noticeable"* you would see it on a normal viewing
- *"obvious"* it dominates the clip; a viewer would reject it

If you genuinely find no defect for a property, emit nothing for it. An empty list is a valid and meaningful answer — do not invent defects to fill the list.

**Step 3.** "defect_summary": one short phrase naming the worst defect you listed, or "none" if the list is empty.

**Step 4.** "reasoning": one or two sentences on why you assigned those obviousness levels.

**Return only:**

```
{"observation":"",
 "defects":[{"property":"appeal|plausibility|coherence","what":"",
             "obviousness":"subtle|noticeable|obvious"}],
 "defect_summary":"",
 "reasoning":""}
```