REBIND: MULTI-REFERENCE VIDEO EDITING VIA STRUCTURED INSTRUCTIONS WITH EXPLICIT REFERENCE RELATIONSHIPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reference video editing integrates content from multiple reference images while preserving relevant content from a source video. However, operation-centric instructions often leave preservation requirements and content-source relationships underspecified. Even detailed instructions may associate entities or attributes with the wrong visual sources, introducing inconsistent training supervision. To address these limitations, we introduce ReBind, a supervision framework that combines comprehensive editing and preservation semantics with accurate source attribution. As it's core, ReBind-Instruct generates comprehensive, reference-consistent instructions through a two-stage training process. To learn these capabilities, Structured Instruction Learning establishes instruction generation through supervised fine-tuning, while Reference Attribution Optimization applies GRPO with content-level and relation-level rewards to improve semantic completeness and correctness and penalize source assignment errors. The resulting annotations provide scalable supervision for ReBind-Edit, which extends a pretrained text-to-video model to multi-reference video editing through joint visual conditioning on the source video and reference images. Controlled ablations show that corrupting source assignments degrades editing performance despite largely retained instruction quality, demonstrating the value of accurate attribution. ReBind-Instruct achieves the highest combined Instruction Quality and Reference Accuracy score among evaluated MLLMs, while ReBind-Edit achieves state-of-the-art open-source performance on IntelligentVBench and UniVBench across single- and multi-reference editing. We will publicly release both models.

1 INTRODUCTION

Recent advances in diffusion-based video generation (Ho et al., 2022; Peebles & Xie, 2023; Wang et al., 2025; HaCohen et al., 2026; Kong et al., 2024; Wei et al., 2026b) have enabled increasingly controllable reference-image-conditioned video editing (Jiang et al., 2025; Ye et al., 2025; Lin et al., 2026; Chen et al., 2026a). Given a source video and one or more reference images, an editor aims to introduce designated identities and visual attributes while preserving the content and dynamics of the source video. In multi-reference image-conditioned video editing (MRVE), operation-centric instructions commonly used in large-scale editing datasets (Wu et al., 2025b; Bai et al., 2025a; He et al., 2025) primarily describe the requested modifications. The broader editing and preservation semantics, together with their relationships to the visual sources, remain only partially specified.

Prior work highlights the role of high-quality supervision and visually grounded instructions in video editing (Du et al., 2025). Multi-reference editing additionally requires accurate source attribution: a character's identity, clothing, and motion may come from different references and the source video (Figure 1). Constructing reliable supervision therefore requires comparing source–target video pairs and reference images to identify edited and preserved content and associate each entity and attribute with its intended source. Even detailed, semantically accurate descriptions can contain incorrect source assignments, making the instructions inconsistent with the target edits. These attribution errors motivate training an instructor to generate comprehensive editing and preservation instructions with accurate entity- and attribute-level source correspondence.

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.

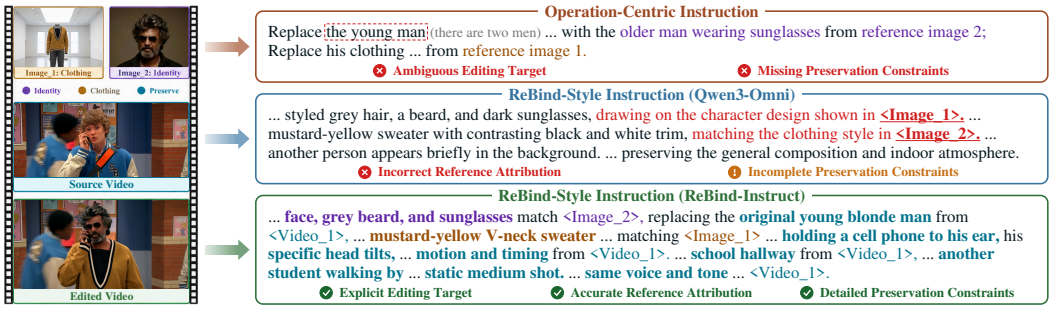

Figure 1: Comparison between an operation-centric instruction and ReBind-style instructions generated by Qwen3-Omni and ReBind-Instruct.

We introduce **ReBind**, a supervision framework that learns to specify editing and preservation semantics together with accurate source attribution. **ReBind instructions** state what content to edit and preserve and associate the relevant entities and attributes with their intended visual sources (Figure 1). Inline reference identifiers provide a standardized notation for expressing these assignments.

To generate this supervision, we develop **ReBind-Instruct**, a specialized MLLM trained on source–edited video pairs and their reference images. **Structured Instruction Learning** establishes ReBind instruction generation through supervised fine-tuning. **Reference Attribution Optimization** then applies GRPO with complementary objectives for the completeness and correctness of edited and preserved content, and for relation-level source attribution. The attribution objective requires semantically matching entity descriptions to also agree on their source identifiers, directly penalizing cross-reference assignment errors. We apply ReBind-Instruct across the editing corpus to construct scalable ReBind supervision for training **ReBind-Edit**. Through a lightweight extension of the model's input formulation, ReBind-Edit accepts a source video and a variable number of reference images as visual conditions. Guided by the explicit source-content relationships in its training supervision, the editor learns to incorporate designated reference content while preserving the intended content and dynamics of the source video.

Controlled editor experiments demonstrate the effectiveness of ReBind supervision under the same architecture and training configuration, with ReBind-Instruct providing further gains (Table 4). Instructor ablations examine how content-level and relation-level objectives affect IQ and RA (Table 5). Beyond these analyses, ReBind-Instruct achieves the best overall instruction-generation performance among the compared MLLMs, while ReBind-Edit substantially outperforms state-of-the-art open-source video editing models on IntelligentVBench (Pan et al., 2026) and UniVBench (Wei et al., 2026a). Our contributions are summarized as follows:

- **Conceptually,** we introduce ReBind, a supervision framework that learns accurate source attribution alongside comprehensive editing and preservation semantics. It addresses incorrect entity and attribute source assignments that can persist even in detailed instructions.
- **Methodologically,** we develop ReBind-Instruct with content-level and relation-level training objectives to generate these annotations at scale, and use them to train ReBind-Edit for reference-guided modification and source-video preservation.
- **Empirically,** ReBind-Instruct achieves the best overall instruction quality among evaluated MLLMs, and ReBind-Edit achieves state-of-the-art open-source performance on IntelligentVBench and UniVBench.

2 Related Work

2.1 Multimodal Instruction Generation

Video captioning methods have made substantial progress in producing detailed and accurate descriptions (Shi et al., 2025; Sun et al., 2024; Chai et al., 2025; Zhang et al., 2026). ShareGPT4Video and LLaVA-Video further scale video instruction tuning through dense captions and synthetic instruction data (Chen et al., 2024; Zhang et al., 2024). Recent studies extend instruction generation toward generation-oriented supervision (Du et al., 2025), audio-visual descriptions (Xu et al., 2025b;

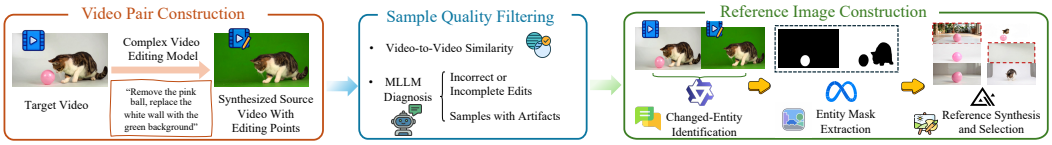

Figure 2: Construction pipeline for reference-image-conditioned video editing. We generate source videos via inverse editing, filter pairs using video similarity and MLLM-based edit verification, and construct reference images for identified changes.

Chen et al., 2025b), and comparative understanding of video pairs (Wu et al., 2025a). MRVE instruction generation introduces an additional requirement: identifying edited and preserved content while consistently associating these semantics with the source video and individual references.

Preference optimization and reinforcement learning enable task-specific post-training beyond supervised learning (Shao et al., 2024). Video-SALMONN 2 (Tang et al., 2025) applies multi-round DPO to audio-visual reasoning, while DiaDem adopts two-stage GRPO for dialogue description (Rafailov et al., 2023; Chen et al., 2026b). VideoCap-R1 and AVoCaDO design GRPO rewards for action captioning and audio-visual temporal coherence (Meng et al., 2025; Chen et al., 2025b). ReBind-Instruct instead optimizes semantic completeness and relation-level reference attribution for comprehensive, reference-consistent MRVE instructions.

2.2 Reference-Conditioned Video Editing

Text-guided video editing (Wu et al., 2025b; Qin et al., 2024; Yang et al., 2025; Zhou et al., 2026; Cong et al., 2026) enables semantic manipulation, but text alone provides limited control over identity, appearance, and localized visual attributes. Reference-conditioned video editing methods (Ye et al., 2025; Liu et al., 2025a; Jiang et al., 2025; Wu et al., 2026; Liu et al., 2025b; 2026b) address this limitation by introducing visual exemplars that improve appearance fidelity and editing precision. Recently, unified video models (Pan et al., 2025; Wei et al., 2025; Chen et al., 2025a) integrate video understanding, generation, and editing within a shared framework, broadening the scope of reference-conditioned manipulation. Among these models, Bernini (Liu et al., 2026a) combines MLLM-based semantic planning with a DiT renderer and demonstrates complex instruction-guided editing while preserving source-video details. This makes Bernini suitable for constructing source videos through inverse editing, but its reference-guided editor accepts only one reference image and does not address multi-reference coordination.

3 Dataset Construction

We construct a synthetic MRVE dataset comprising source–target video pairs and multiple reference images. Using target videos from OpenVid-1M (Nan et al., 2025) and AVSET-10M (Cheng et al., 2024), we apply Bernini (Liu et al., 2026a) with inverse instructions to generate source videos, as illustrated in Figure 2. We filter the resulting pairs using video similarity and MLLM-based edit verification. To construct reference images, Qwen3-Omni-30B-A3B (Bai et al., 2025b) identifies entities that differ between the source and target videos, and SAM3 (Carion et al., 2025) extracts their multi-frame observations. Rather than directly using target-video crops, we employ Flux2 (Labs et al., 2025) to synthesize identity- or attribute-preserving reference candidates with varied viewpoints, poses, backgrounds, and illumination. We retain candidates that preserve the intended entity semantics while remaining visually distinct from the target frames. The final dataset contains approximately 100K editing triplets, comprising 52.3% single-reference and 47.7% multi-reference samples with 1–5 reference images per sample. ReBind instruction annotations are described below; further processing details appear in Appendix A.

4 Method

We train ReBind-Instruct through Structured Instruction Learning and Reference Attribution Optimization. Its instructions provide ReBind-Edit with explicit entity- and attribute-level source assignments during training, specifying what to transfer from each reference and what to preserve from the source video.

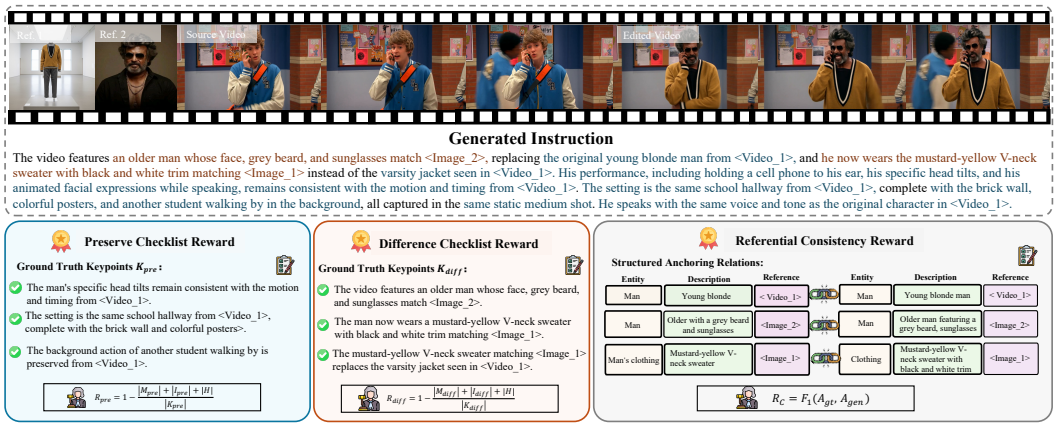

Figure 3: Illustration of our dual checklist rewards for content completeness and referential consistency reward for accurate reference attribution.

4.1 Stage 1: Structured Instruction Learning

Training Dataset. We use Gemini 3.1 Pro to annotate source–edited video pairs and their reference images with ReBind instructions. We first apply rule-based filtering for length, repetition, and formatting, followed by prompt-based quality verification with Gemini, retaining approximately 40K SFT examples (Appendix A.3.1).

Training Objective. We initialize ReBind-Instruct from Qwen3-Omni-30B-A3B (Xu et al., 2025b) and fine-tune it to predict these annotations from the video pairs and reference images. This stage teaches the model to describe edited and preserved content and express entity- and attribute-level source assignments using inline reference identifiers.

4.2 Stage 2: Reference Attribution Optimization

While SFT provides a strong initialization, it cannot directly optimize for task-specific objectives such as semantic completeness and reference attribution accuracy. To address this, we apply GRPO (Shao et al., 2024) to further improve ReBind-Instruct by optimizing a composite reward function. We describe the training data preparation and reward design below; the GRPO formulation is provided in Appendix B.3.

Training Dataset. For the Reference Attribution Optimization stage, we randomly sample approximately 2K video editing samples from the constructed dataset and manually annotate them with ground-truth ReBind instructions. To enable efficient reward evaluation during GRPO training, we perform offline extraction of reward components from human-annotated instructions using GPT-5.1, including dual checklist keypoints and relation triples.

4.2.1 Reward Design

ReBind instruction optimization targets semantic completeness, reference attribution, and generation stability. We combine three rewards: (1) dual checklist rewards $\mathcal{R}_D$ that evaluate omissions, inaccuracies, and hallucinations in preserved and edited content; (2) referential consistency reward $\mathcal{R}_C$ based on entity–description–reference triple matching; and (3) length reward $\mathcal{R}_L$ that regulates repetition and length. Figure 3 visualizes $\mathcal{R}_D$ and $\mathcal{R}_C$; all components are detailed in subsequent subsections.

4.2.2 Dual Checklist Reward

To enhance instruction completeness and accuracy, we propose dual checklist rewards that evaluate content coverage through missing, incorrect, and hallucination analysis. The dual design separately considers content preserved from the source video and content modified using reference images. Preserved-content keypoints cover visual and temporal elements retained from the source video

<Video_1>, including subject motion, camera behavior, background context, and unaffected entities. In contrast, edited-content keypoints cover modifications realized using reference images <Image_X>, such as changes in subject identity, appearance, clothing, or background.

From each annotated ground-truth instruction S_{gt}, we derive two distinct keypoint sets: preserved-content keypoints $\mathcal{K}_{pre}$ and edited-content keypoints $\mathcal{K}_{diff}$. Given a generated instruction S_{gen}, GPT-5.1 jointly evaluates it against $\mathcal{K}_{pre}$ and $\mathcal{K}_{diff}$. Missing keypoints are checklist items not covered by S_{gen}, while incorrect keypoints are mentioned but inaccurately described. We denote the corresponding errors by $\mathcal{M}_{pre}$, $\mathcal{I}_{pre}$, $\mathcal{M}_{diff}$, and $\mathcal{I}_{diff}$. The evaluator also identifies a shared set of hallucinated descriptions $\mathcal{H}$ by considering both checklists jointly. We compute the two checklist rewards as

$$\mathcal{R}_{pre} = 1 - \frac{|\mathcal{M}_{pre}| + |\mathcal{I}_{pre}| + |\mathcal{H}|}{|\mathcal{K}_{pre}|}. \tag{1}$$

$$\mathcal{R}_{diff} = 1 - \frac{|\mathcal{M}_{diff}| + |\mathcal{I}_{diff}| + |\mathcal{H}|}{|\mathcal{K}_{diff}|}. \tag{2}$$

Here, $\mathcal{R}_{pre}$ and $\mathcal{R}_{diff}$ evaluate the semantic coverage and correctness of preserved and edited content, respectively. The final dual checklist reward is $\mathcal{R}_D = \frac{1}{2}(\mathcal{R}_{pre} + \mathcal{R}_{diff})$.

4.2.3 Referential Consistency Reward

Checklist-based rewards evaluate keypoints holistically and therefore do not explicitly isolate whether an otherwise correct entity description is assigned to the wrong visual source. The referential consistency reward complements this content-level evaluation with a dedicated relation-level signal for *cross-reference confusion*. To explicitly evaluate source attribution, we introduce a referential consistency reward $\mathcal{R}_C$ based on structured entity–description–reference triples. Each relation is formulated as $a_i = \langle e_i, d_i, r_i \rangle$, comprising an entity identifier, visual description, and inline reference token. By requiring agreement across all three components, $\mathcal{R}_C$ penalizes incorrect source assignments even when the entity and description are otherwise correct. Ground-truth and generated relations form indexed collections $\mathcal{A}_{gt} = (a_1^{gt}, \ldots, a_M^{gt})$ and $\mathcal{A}_{gen} = (a_1^{gen}, \ldots, a_N^{gen})$, respectively.

We employ GPT-5.1 to extract and semantically match triples from ground-truth and generated instructions, then construct a binary matching matrix $W \in \{0,1\}^{M \times N}$:

$$w_{i,j} = \begin{cases} 1, & \text{if } e_i^{gt} \equiv e_j^{gen},\ d_i^{gt} \equiv d_j^{gen},\ r_i^{gt} = r_j^{gen} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\equiv$ denotes semantic equivalence determined by the evaluator. A match permits paraphrased entity descriptions but requires exact reference-token agreement, assigning zero score when otherwise correct content is attributed to the wrong reference. We compute a maximum-cardinality bipartite matching using edges with $w_{i,j} = 1$. Each triple participates in at most one matched pair, without any ordering constraint. Let K denote the number of matched pairs. Precision and recall are

$$\text{Precision} = \frac{K}{N}, \qquad \text{Recall} = \frac{K}{M}. \tag{4}$$

The referential consistency reward combines these components via F1-score:

$$\mathcal{R}_C = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2K}{M + N}. \tag{5}$$

We set precision or recall to zero when its denominator is zero, and $\mathcal{R}_C = 0$ when $K = 0$. Repeated generated entries remain in N but cannot reuse a ground-truth triple; permuting either collection leaves K unchanged.

4.2.4 Length Reward

To mitigate repetition-induced verbosity, we introduce a length reward that progressively penalizes outputs exceeding a target length. Following (Chen et al., 2025b), we adopt a unidirectional penalty structure. The thresholds $\tau_1 = 512$ and $\tau_2 = 1024$ are set based on empirical analysis, where τ_1 is

approximately the longest instruction in the ground-truth distribution and τ_2 is approximately twice the maximum ground-truth length:

$$\mathcal{R}_L = \begin{cases} 1.0, & \text{if } L < \tau_1 \\ 1 - \frac{L-\tau_1}{\tau_2-\tau_1}, & \text{if } \tau_1 \le L < \tau_2 \\ 0.0, & \text{otherwise} \end{cases} \tag{6}$$

where L denotes the token count of the generated instruction. We combine the three reward components to obtain the final reward:

$$R(S_g) = \mathcal{R}_D(S_g) + \mathcal{R}_C(S_g) + \mathcal{R}_L(S_g). \tag{7}$$

4.3 Reference-Conditioned Video Editor

Architecture. ReBind-Edit adapts the pretrained LTX-2.3 text-to-video model (HaCohen et al., 2026) to multi-reference video editing with ReBind supervision. Given a source video and K reference images, we construct a unified latent sequence $z = [z_{ref_1}, \ldots, z_{ref_K}, z_{src}, z_{edit}]$, where the reference order is consistent with the inline identifiers in the instruction. The source and reference latents serve as clean conditions, while the edited-video latent follows the denoising trajectory (Team et al., 2025). ReBind instructions are encoded through the model's cross-attention pathway, with inline identifiers explicitly expressing the intended source–content relationships. These associations are learned from paired instructions and visual inputs during training.

Training. We train ReBind-Edit on tuples $(V_{src}, V_{edit}, \{I_1, \ldots, I_K\}, \hat{S})$, where $\hat{S}$ is generated by ReBind-Instruct. During training, only the edited-video latent is perturbed. Given a noise level $\sigma_t \in [0, 1]$ and $\epsilon \sim \mathcal{N}(0, I)$, we define

$$z_{edit,t} = (1 - \sigma_t) z_{edit} + \sigma_t \epsilon, \qquad u_{edit} = \epsilon - z_{edit}, \tag{8}$$

where u_{edit} is the target velocity, and form $\tilde{z}_t = [z_{ref_1}, \ldots, z_{ref_K}, z_{src}, z_{edit,t}]$. Let $\boldsymbol{\tau}_t$ denote the per-token timesteps, assigning σ_t to edited-video tokens and zero to clean condition tokens. We optimize the flow-matching objective over the edited-video positions:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{z,\sigma_t,\epsilon}\left[\left\| u_{edit} - v_\theta(\tilde{z}_t, \boldsymbol{\tau}_t, \hat{S})_{edit} \right\|_2^2\right], \tag{9}$$

where $v_\theta(\cdot)_{edit}$ is the predicted velocity at edited-video positions, conditioned on the visual inputs and $\hat{S}$.

Inference. At inference, Qwen3-Omni-30B-A3B (Xu et al., 2025b) converts the original instruction S into a ReBind-format instruction $\hat{S}$ conditioned on V_{src} and $\{I_1, \ldots, I_K\}$. ReBind-Edit then predicts $\widehat{V}_{edit}$ from $(V_{src}, \{I_1, \ldots, I_K\}, \hat{S})$, without access to V_{edit}.

5 Experiments

We evaluate ReBind at the instruction and video levels. We compare ReBind-Instruct with general-purpose MLLMs, isolate the effect of ReBind instructions under a fixed editor configuration, and benchmark ReBind-Edit against state-of-the-art single- and multi-reference editing methods.

5.1 Evaluation Setup

Benchmarks. We evaluate video editing on IntelligentVBench (TIV2V) (Pan et al., 2026) and the single- and multi-reference subsets of UniVBench (Wei et al., 2026a). For instruction generation, we use the multi-reference portion of a 1,200-case human-annotated UniVBench-derived evaluation set. Benchmark details and evaluation-set construction are provided in Appendices C.1 and C.2.

Metrics. We assess instructions using Instruction Quality (IQ), measuring semantic completeness and correctness, and Reference Accuracy (RA), measuring source attribution through semantic entity–description matching with exact reference-token agreement. RA is computed as maximum-cardinality one-to-one matching F1, and Overall averages IQ and RA. For video editing, we retain each benchmark's metrics. GPT-5.1 evaluates instructions and UniVBench edits, while IntelligentVBench uses Gemini-2.5-Pro. Metric definitions and aggregation rules are detailed in Appendix C.1.

Table 1: Single-reference (RVE) and multi-reference (MRVE) image-conditioned video editing on UniVBench. Scores range from 0 to 10.

Single Reference↑									
Method	**Sub.**	**Back.**	**Action**	**Cam.**	**Color**	**Light.**	**Style**	**Rel.Pos.**	**Overall**
Closed-Source Models									
Kling-o3 (2026-03)	6.42	7.94	7.51	8.61	7.36	7.85	6.65	7.65	7.50
Open-Source Models									
UniVideo(query)	3.08	3.51	5.97	8.03	5.04	5.93	4.02	6.92	5.31
UniVideo(hidden)	5.15	6.82	6.02	8.03	6.36	7.12	5.58	6.76	6.48
Kiwi-Edit	3.14	3.38	6.23	7.96	4.80	5.86	4.08	7.07	5.32
OmniWeaving	3.08	2.95	6.04	7.67	4.22	5.46	3.68	6.82	4.99
ReBind-Edit (Ours)	**5.61**	**7.33**	**6.72**	**8.11**	**6.76**	**7.45**	**5.78**	**7.34**	**6.89**

Multi Reference↑									
Method	**Sub.**	**Back.**	**Action**	**Cam.**	**Color**	**Light.**	**Style**	**Rel.Pos.**	**Overall**
Closed-Source Models									
Kling-o3 (2026-03)	5.99	6.45	6.84	8.01	6.96	7.37	5.85	7.07	6.82
Open-Source Models									
UniVideo(query)	4.61	4.75	5.90	**7.80**	5.73	6.22	4.83	6.39	5.78
UniVideo(hidden)	4.57	4.84	5.67	7.64	5.89	6.44	4.96	6.49	5.81
Kiwi-Edit	4.18	4.67	6.05	7.79	5.63	6.25	4.81	6.65	5.75
OmniWeaving	4.28	4.00	5.65	7.34	5.32	5.76	4.29	6.02	5.33
ReBind-Edit (Ours)	**5.66**	**5.93**	**6.12**	7.71	**6.59**	**6.95**	**5.36**	**6.97**	**6.41**

Baselines. For instruction generation, we compare ReBind-Instruct with the Gemini (Comanici et al., 2025; Google, 2026), Qwen-Omni (Xu et al., 2025a;b), and Gemma (Google DeepMind, 2025) families, including both Captioner and Instruct variants of Qwen3-Omni. For video editing, we compare ReBind-Edit with UniVideo (Wei et al., 2025), VACE (Jiang et al., 2025), VINO (Chen et al., 2026a), Kiwi-Edit (Lin et al., 2026), and OmniWeaving (Pan et al., 2026), with HappyHorse 1.0 (HappyHorse, 2026) and Kling-o3 (Kuaishou Technology, 2024) as closed-source references. The benchmark-specific editing baselines are listed in Tables 2 and 1; instruction-generation baselines are listed in Table 3. Model variants and inference protocols are detailed in Appendix C.3.

Evaluation Protocol. All instruction-generation models receive identical source videos, edited videos, and reference images. For video editing, all methods start from the same source videos, reference images, and original benchmark instructions. ReBind-Edit uses Qwen3-Omni to rewrite instructions without access to target videos; each editing baseline follows its official preprocessing and inference configuration.

5.2 Reference-Conditioned Video Editing

Table 1 reports single- and multi-reference results on UniVBench. ReBind-Edit achieves the best open-source overall performance in both settings, surpassing the hidden-state UniVideo variant by 0.41 and 0.60, respectively.

Table 2 reports single-reference results on IntelligentVBench. ReBind-Edit achieves the best average performance among all evaluated methods, surpassing Kling-o3 by 0.17, HappyHorse 1.0 by 0.64, and the strongest open-source baseline OmniWeaving by 0.49. Additional single-reference qualitative comparisons are provided in Appendix D.3.

Table 2: Single-reference video editing on IntelligentVBench.

Method	**Single Reference↑**				
	IF	**CP**	**VQ**	**Min**	**Avg**
Closed-Source Models					
Kling-o3 (2026-03)	4.60	4.13	3.91	3.60	4.21
HappyHorse 1.0	2.93	4.41	3.89	2.70	3.74
Open-Source Models					
VACE-Wan2.1	1.46	1.42	1.71	1.27	1.53
VACE-LTX	1.43	1.36	1.25	1.20	1.35
VINO	2.86	2.90	2.52	2.11	2.76
UniVideo(query)	3.22	3.91	3.26	2.66	3.46
UniVideo(hidden)	3.13	4.01	2.93	2.56	3.36
OmniWeaving	4.00	4.04	3.65	3.31	3.89
ReBind-Edit (Ours)	**4.59**	**4.41**	**4.16**	**3.94**	**4.38**

Figure 5 shows cross-reference confusion in OmniWeaving, weak reference consistency in Kiwi-Edit, and temporal inconsistencies in UniVideo.

Some closed-source outputs also show imperfect identity preservation, whereas ReBind-Edit better maintains source attribution and temporal consistency.

5.3 User Study

For 50 randomly sampled UniVBench cases, 15 participants view the source video, reference images, and editing instruction, then rank anonymized outputs from ReBind-Edit, UniVideo, Kiwi-Edit, and OmniWeaving. Figure 4 shows the highest normalized preference scores for ReBind-Edit in instruction following, source-video preservation, reference-image preservation, and visual quality. The largest margin over the strongest baseline is in reference-image preservation (87.3 vs. 49.7).

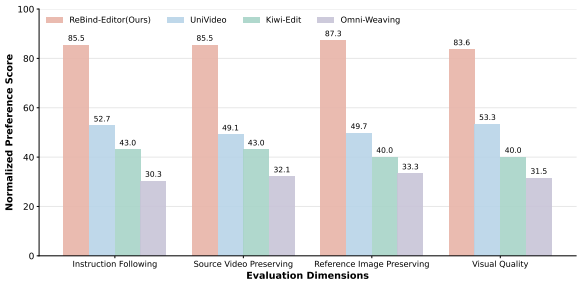

Figure 4: Human evaluation on UniVBench. Normalized preference scores; higher is better.

5.4 ReBind Instruction Generation Quality

Table 3 evaluates ReBind-Instruct against general-purpose MLLMs on the annotated UniVBench multi-reference subset. ReBind-Instruct achieves the highest Overall score among all evaluated models, including closed-source systems. It surpasses the strongest open-source baseline Gemma4-31B-it by 17.54 and Gemini-3.1-Pro by 4.57 in Overall. These results demonstrate stronger overall MRVE instruction generation, as jointly measured by instruction quality and reference attribution.

Table 3: ReBind instruction generation on the annotated UniVBench multi-reference subset. Scores are percentages.

Model	Size	IQ↑	RA↑	Overall↑
Closed-Source Models				
Gemini-2.5-Pro	-	46.37	63.85	55.11
Gemini-3.1-Pro	-	59.24	52.90	56.07
Open-Source Models				
Qwen2.5-Omni-3B	3B	10.10	21.86	15.98
Qwen2.5-Omni-7B	7B	14.32	35.37	24.84
Qwen3-Omni-Captioner	30B-A3B	22.01	39.47	30.74
Qwen3-Omni-Instruct	30B-A3B	31.82	49.59	40.70
Gemma4-31B-it	31B	49.49	36.71	43.10
ReBind-Instruct (Ours)	30B-A3B	**61.68**	**59.60**	**60.64**

5.5 Ablation Study

5.5.1 Effect of Instruction Supervision on Editor Learning

Table 4: Effect of instruction supervision on MRVE on the UniVBench multi-reference subset.

Instruction supervision	Instructor	Instruction (%)		Video editing								
		IQ↑	RA↑	Sub.	Back.	Act.	Cam.	Col.	Light.	Style	Rel.	Overall↑
Operation-centric	Qwen3-Omni	20.41	45.37	3.85	4.08	4.92	6.95	4.96	5.62	3.94	5.82	5.02
ReBind	Qwen3-Omni	31.82	49.59	4.36	4.41	5.38	7.21	5.51	6.18	4.44	6.27	5.47
ReBind with corrupted attribution	ReBind-Instruct	60.62	43.62	4.72	4.90	5.57	7.34	5.79	6.38	4.75	6.47	5.74
ReBind	**ReBind-Instruct**	**61.68**	**59.60**	**5.08**	**5.38**	**5.76**	**7.46**	**6.07**	**6.57**	**5.06**	**6.67**	**6.01**

Table 4 compares four supervision variants with the same 50K training cases, editor architecture, and 5K-step budget. With Qwen3-Omni, ReBind improves editing Overall from 5.02 to 5.47. ReBind-Instruct raises it to 6.01 with higher IQ and RA under the same instruction format.

For attribution control, an MLLM (Comanici et al., 2025) evaluates source assignments in a subset of annotations against the source video, target video, and reference images. We redirect only image reference assignments judged correct to alternatives judged incompatible with their descriptions. All other assignments, including those to the source video, remain unchanged. The modified annotations train the editor with descriptive text, format, and visual data preserved.



Figure 5: Qualitative comparison of multi-reference video editing on UniVBench.

Corruption reduces RA from 59.60 to 43.62 with only a slight IQ decrease. Editing Overall drops by 0.27 points, with declines across all eight dimensions, supporting the importance of accurate source attribution for editor learning.

5.5.2 Impact of Different Reward Components

Table 5 examines the training stages and reward components. Stage 1 supervised fine-tuning improves Overall by 14.28 points, followed by a further gain of 4.00 points from GRPO with $\mathcal{R}_D$. This improvement is primarily reflected in IQ, consistent with the content reward's focus on semantic completeness and correctness. Adding $\mathcal{R}_C$ improves RA and increases Overall by 1.42 points. This gain highlights the value of explicitly optimizing source attribution alongside semantic coverage. Incorporating $\mathcal{R}_L$ provides another 0.24 points, with the full objective achieving the highest Overall of 60.64.

Table 5: Training-stage and reward component ablations.

Model	Reward			Metrics (%)		
	$\mathcal{R}_D$	$\mathcal{R}_C$	$\mathcal{R}_L$	IQ↑	RA↑	Overall↑
Qwen3-Omni Instruct	–	–	–	31.82	49.59	40.70
Ours-Stage1	–	–	–	56.84	53.11	54.98
Ours-Stage2	✓	–	–	63.81	54.15	58.98
	✓	✓	–	60.74	60.07	60.40
	✓	✓	✓	61.68	59.60	60.64

6 Conclusion

We presented ReBind, a supervision framework that jointly specifies editing and preservation requirements with accurate source attribution. ReBind-Instruct learns to generate these annotations through supervised instruction learning and dedicated attribution optimization. Its supervision supports ReBind-Edit, which extends a pretrained text-to-video model to multi-reference video editing. ReBind-Instruct achieves the highest combined IQ and RA score among evaluated MLLMs, while ReBind-Edit achieves state-of-the-art open-source performance on IntelligentVBench and UniVBench. Controlled ablations show that attribution errors impair editing despite detailed instructions, highlighting the importance of accurate source attribution for editor learning.

AI USE STATEMENT

Generative AI tools were used to support synthetic dataset creation, method implementation, proof writing, translation, and dataset cleaning and reformatting. We also used these tools to improve the manuscript's readability and to suggest candidate titles and keywords.

We did not use generative AI to develop theoretical models or conceptual frameworks, formulate mathematical claims, supply essential components of proofs, propose or refine hypotheses, design or provide feedback on research methods or experiments, perform qualitative or thematic analyses, or interpret research findings.

All AI-assisted material was reviewed by the authors. Generated code was verified, and mathematical derivations were checked for logical soundness. Synthetic and reformatted datasets were manually audited through random sampling, while translations and edited passages were reviewed to ensure that they retained the intended technical meaning. The authors take full responsibility for the final manuscript and its associated claims and artifacts, including all content produced with AI assistance.

ETHICS STATEMENT

This work constructs synthetic video-editing data from OpenVid-1M and AVSET-10M using pretrained models and includes human instruction annotation and user evaluation. Data use and distribution should respect applicable licenses and privacy requirements. As with other video editing methods, outputs may reflect biases in the underlying data and models. We encourage responsible use and clear disclosure of edited content where appropriate.

REFERENCES

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025a.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 16845–16889, 2025.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

Junyi Chen, Tong He, Zhoujie Fu, Pengfei Wan, Kun Gai, and Weicai Ye. Vino: A unified visual generator with interleaved omnimodal context. *arXiv preprint arXiv:2601.02358*, 2026a.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024.

Xinlong Chen, Yue Ding, Weihong Lin, et al. Avocado: An audiovisual video captioner driven by temporal orchestration. *arXiv preprint arXiv:2510.10395*, 2025b.

Xinlong Chen, Weihong Lin, Jingyun Hua, Liang Wang, Tieniu Tan, et al. Diadem: Advancing dialogue descriptions in audiovisual video captioning for multimodal large language models. *arXiv preprint arXiv:2601.19267*, 2026b.

Xize Cheng, Ziang Zhang, Minghui Fang, Zehan Wang, Ruofan Hu, Siqi Zheng, Rongjie Huang, Tao Jin, and Zhou Zhao. Avset-10m: An open large-scale audio-visual dataset with high correspondence. 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34364–34374, 2026.

Yang Du, Zhuoran Lin, Kaiqiang Song, Biao Wang, Zhicheng Zheng, Tiezheng Ge, Bo Zheng, and Qin Jin. Vc4vg: Optimizing video captions for text-to-video generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1124–1138, 2025.

Google. Gemini api: Gemini 3.1 pro. `https://ai.google.dev/gemini-api`, 2026. Version 3.1 Pro.

Google DeepMind. Gemma 4: Multimodal open models. `https://ai.google.dev/gemma/docs`, 2025. Version 31B.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

HappyHorse. HappyHorse: Ai video generation platform. `https://happyhorses.io/`, 2026. Accessed: 2026-06-23.

Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing. *arXiv preprint arXiv:2512.07826*, 2025.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

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

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

Kuaishou Technology. Kling ai video generation platform. `https://klingai.com`, 2024. Accessed: 2026-07-01.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Yiqi Lin, Guoqiang Liang, Ziyun Zeng, Zechen Bai, Yanzhe Chen, and Mike Zheng Shou. Kiwi-edit: Versatile video editing via instruction and reference guidance. *arXiv preprint arXiv:2603.02175*, 2026.

Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. Stablev2v: Stabilizing shape consistency in video-to-video editing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025a.

Chenchen Liu, Junyi Chen, Lei Li, Lu Chi, Mingzhen Sun, Zhuoying Li, Yi Fu, Ruoyu Guo, Yiheng Wu, Ge Bai, and Zehuan Yuan. Bernini: Latent semantic planning for video diffusion. *arXiv preprint arXiv:2605.22344*, 2026a.

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. *arXiv preprint arXiv:2606.08260*, 2026b.

Xinyu Liu, Hangjie Yuan, Yujie Wei, Jiazheng Xing, Yujin Han, Jiahao Pan, Yanbiao Ma, Chi-Min Chan, Kang Zhao, Shiwei Zhang, et al. Revise: Towards reason-informed video editing in unified models with self-reflective learning. *arXiv preprint arXiv:2512.09924*, 2025b.

Desen Meng, Rui Huang, Zhilin Dai, Xinhao Li, Yifan Xu, Jun Zhang, Zhenpeng Huang, Meng Zhang, Lingshu Zhang, Yi Liu, et al. Videocap-r1: Enhancing mllms for video captioning via structured thinking. *arXiv preprint arXiv:2506.01725*, 2025.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, volume 2025, pp. 1045–1064, 2025.

Kaihang Pan, Qi Tian, Jianwei Zhang, Weijie Kong, Jiangfeng Xiong, Yanxin Long, Shixue Zhang, Haiyi Qiu, Tan Wang, Zheqi Lv, et al. Omniweaving: Towards unified video generation with free-form composition and reasoning. *arXiv preprint arXiv:2603.24458*, 2026.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Bosheng Qin, Juncheng Li, Siliang Tang, Tat-Seng Chua, and Yueting Zhuang. Instructvid2vid: Controllable video editing with natural language instructions. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Seedance Team. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yang Shi, Jiaheng Liu, Yushuo Guan, Zhenhua Wu, Yuanxing Zhang, Zihao Wang, Weihong Lin, Jingyun Hua, Zekun Wang, Xinlong Chen, et al. Mavors: Multi-granularity video representation for multimodal large language model. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10994–11003, 2025.

Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. video-salmonn: Speech-enhanced audio-visual large language models. *arXiv preprint arXiv:2406.15704*, 2024.

Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-salmonn 2: Caption-enhanced audio-visual large language models. *arXiv preprint arXiv:2506.15220*, 2025.

Kling Team, Jialu Chen, Yuanzheng Ci, Xiangyu Du, Zipeng Feng, Kun Gai, Sainan Guo, Feng Han, Jingbin He, Kang He, et al. Kling-omni technical report. *arXiv preprint arXiv:2512.16776*, 2025.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 3(4):6, 2025.

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. *arXiv preprint arXiv:2510.08377*, 2025.

Jianhui Wei, Xiaotian Zhang, Yichen Li, Yuan Wang, Yan Zhang, Ziyi Chen, Zhihang Tang, Wei Xu, and Zuozhu Liu. Univbench: Towards unified evaluation for video foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25654–25666, 2026a.

Yujie Wei, Xinyu Liu, Shiwei Zhang, Hangjie Yuan, Jinbo Xing, Zhekai Chen, Xiang Wang, Haonan Qiu, Rui Zhao, Yutong Feng, et al. Dreamvideo-omni: Omni-motion controlled multi-subject video customization with latent identity reinforcement learning. *arXiv preprint arXiv:2603.12257*, 2026b.

Jiangtao Wu, Shihao Li, Zhaozhou Bian, Jialu Chen, Runzhe Wen, An Ping, Yiwen He, Jiakai Wang, Yuanxing Zhang, and Jiaheng Liu. Vidic: Video difference captioning. *arXiv preprint arXiv:2512.03405*, 2025a.

Jianzong Wu, Hao Lian, Jiongfan Yang, Dachao Hao, Ye Tian, Yunhai Tong, Jingyuan Zhu, Biaolong Chen, Qiaosong Qi, Aixi Zhang, et al. Loomvideo: Unifying multimodal inputs into video generation and editing. *arXiv preprint arXiv:2606.06042*, 2026.

Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16692–16701, 2025b.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025a. URL https://arxiv.org/abs/2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025b.

Xiangpeng Yang, Linchao Zhu, Hehe Fan, and Yi Yang. Videograin: Modulating space-time attention for multi-grained video editing. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zixuan Ye, Xuanhua He, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Qifeng Chen, and Wenhan Luo. Unic: Unified in-context video editing. *arXiv preprint arXiv:2506.04216*, 2025.

Yabo Zhang, Kunchang Li, Dewei Zhou, Xinyu Huang, and Xun Wang. Images in sentences: Scaling interleaved instructions for unified visual generation. *arXiv preprint arXiv:2605.12305*, 2026.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Donghao Zhou, Guisheng Liu, Hao Yang, Jiatong Li, Jingyu Lin, Xiaohu Huang, Yichen Liu, Xin Gao, Cunjian Chen, Shilei Wen, et al. Omnishow: Unifying multimodal conditions for human-object interaction video generation. *arXiv preprint arXiv:2604.11804*, 2026.

APPENDIX CONTENTS

A. Dataset Construction and Annotation 14
A.1 Video Pair Construction 14
A.2 Reference Image Synthesis and Selection 15
A.3 Instruction Annotation 15
A.4 Dataset Statistics 16
B. Training and Optimization Details 16
B.1 Training Configuration 16
B.2 Editor Positional Encoding 16
B.3 Complete GRPO Objective 17
C. Evaluation Protocols 18
C.1 Evaluation Benchmarks and Metrics 18
C.2 Instruction-Generation Evaluation Set 20
C.3 Baselines and Inference Protocol 20
C.4 UniVBench Evaluation Prompt 21
D. Additional Results and Analyses 23
D.1 Effect of Instruction Supervision on Editing Behavior 23
D.2 Training Dynamics 23
D.3 Additional Qualitative Results 24
E. Limitations 28

A DATASET CONSTRUCTION AND ANNOTATION

We detail the construction of the training corpus: source–target video pairs (Section A.1), reference images (Section A.2), and instruction annotations (Section A.3). Section A.4 summarizes the resulting dataset. The separate instruction-generation evaluation set is described in Section C.2.

A.1 VIDEO PAIR CONSTRUCTION

We select target videos, specify inverse edits to generate source videos, and filter the resulting pairs. The following subsections describe each step and its selection criteria.

A.1.1 TARGET VIDEO SELECTION CRITERIA

Target videos are collected from OpenVid-1M and AVSET-10M based on content suitability for editing operations. We prioritize videos with clear subjects and stable camera motion to facilitate reliable reverse editing. Specifically, videos are filtered to ensure: (1) minimum resolution of 720P and duration between 5–10 seconds to balance visual quality with computational efficiency, (2) presence of identifiable entities (people, objects, or scenes) suitable for attribute-level editing, and (3) absence of extreme motion blur or occlusion that could compromise editing quality. Videos containing predominantly text overlays or with severe compression artifacts are excluded through automated quality checks.

A.1.2 INVERSE INSTRUCTION DESIGN

To generate source videos through reverse editing, we design inverse instructions that specify attribute transformations opposite to the desired editing direction. For each target video, we first identify editable attributes using automated captioning models, then construct inverse instructions following templates such as "change the [entity] to have [inverse attribute]" or "replace the [entity] with one having [contrasting property]." For example, if the target video shows a person wearing red clothing, the inverse instruction might specify changing to blue clothing. This ensures the generated source video provides a meaningful starting point for forward editing tasks.

A.1.3 QUALITY FILTERING THRESHOLDS

In the video-to-video similarity filtering stage, we compute cosine similarity between CLIP (Radford et al., 2021) embeddings extracted from source and target videos. Pairs with similarity below 0.3

indicate failed edits where the source and target are semantically unrelated, while similarity above 0.95 suggests insufficient editing variation. Both extremes are filtered to ensure the dataset contains meaningful editing challenges. For MLLM verification, GPT-5.1 is prompted to evaluate whether the visual changes between source and target align with the inverse editing instruction. Samples where the model identifies semantic inconsistencies, incomplete edits, or visual artifacts are flagged and excluded. This two-stage process removes approximately 15% of initial pairs.

A.2 Reference Image Synthesis and Selection

Qwen3-Omni-30B-A3B (Xu et al., 2025b) identifies entities that differ between the source and target videos, and SAM3 (Carion et al., 2025) extracts their multi-frame masks and observations. Rather than directly using target-video crops, Flux2 (Labs et al., 2025) uses these observations to synthesize identity- or attribute-preserving reference candidates with varied viewpoints, poses, backgrounds, and illumination. We retain candidates that preserve the intended entity semantics while remaining visually distinct from the target frames, rejecting images that closely reproduce their composition or contain blur, occlusion, truncation, or synthesis artifacts. The highest-quality candidate is selected for each edited entity, yielding one reference image per entity and 2–5 reference images for each multi-reference sample.

A.3 Instruction Annotation

ReBind-Instruct learns from synthetic annotations in Stage 1 and human annotations in Stage 2. After training, it generates ReBind instructions for the editing corpus used to train ReBind-Edit. We describe the preparation of the two instructor-training annotation sets below; model training configurations are given in Section B.1.

A.3.1 Synthetic Annotation and Quality Filtering

For Stage 1, Gemini 3.1 Pro generates ReBind instructions from source–edited video pairs and their reference images. We apply rule-based filtering for length anomalies, repetition, and formatting errors, followed by Gemini-based verification of reference-token consistency, content quality, and format correctness. This process retains approximately 40K SFT examples. The quality-verification prompt is provided below.

Instruction Quality Filtering Prompt

You are a quality control specialist for video editing instructions. Your task is to evaluate whether a generated ReBind instruction meets quality standards for multi-reference video editing.
Input: Source Video Description, Available Reference Tokens (e.g., `<Image_1>`, `<Image_2>`, `<Video_1>`), Generated Instruction, Edit Type.
Evaluation Criteria:
1. Reference Token Consistency: All reference tokens must exist in the available list, follow correct format (`<Image_N>` or `<Video_N>`), and be correctly attributed. No hallucinated or missing tokens.
2. Content Quality: Length should be 50-300 tokens. No excessive repetition (same phrase more than 3 times). Logical flow and clear structure. Covers both preserved and changed elements.
3. Format Correctness: Proper grammar and sentence structure. Clear distinction between preserved and modified elements. No truncated or incomplete sentences.
Decision Rules: FILTER if: reference token errors, severe length anomalies (less than 20 or more than 512 tokens), high repetition, or critical format errors. KEEP if: all tokens are valid, content is coherent and complete.
Evaluate the given instruction and return whether it should be filtered with a brief explanation.

A.3.2 Human Annotation for GRPO Training

To provide high-quality supervision for Stage 2 GRPO training, we recruit eight experienced annotators to manually create ReBind instructions for 2K samples from the synthetic dataset. Annotators are provided with the source video, target video, and reference images, and are tasked with writing comprehensive instructions that: (1) express explicit reference relationships using standardized inline reference tokens (e.g., `<Image_1>`, `<Image_2>`), (2) clearly distinguish between preserved and modified elements, and (3) provide sufficient detail to guide the editing operation. While these

ground-truth instructions are human-annotated, auxiliary training signals such as checklists and reward scores are automatically generated by GPT-5.1 for scalability.

A.4 Dataset Statistics

The final dataset contains approximately 100K video editing triplets with the following distribution: 52.3% single-reference samples (one entity edited) and 47.7% multi-reference samples (2–5 entities edited simultaneously). Videos range from 5–10 seconds in duration with resolutions between 720p and 1080p. Content categories include people (49.2%), animals (12.6%), objects (32.8%), and scenes (5.4%). Each sample contains one reference image per edited entity, yielding 1–5 reference images, together with a ReBind instruction generated through the two-stage ReBind-Instruct pipeline.

B Training and Optimization Details

Section B.1 specifies the training configurations for ReBind-Instruct and ReBind-Edit. Section B.2 details the editor's positional encoding. Section B.3 gives the complete GRPO objective used for Stage 2 Reference Attribution Optimization, complementing the task-specific reward design in Section 4.2.

B.1 Training Configuration

To support reproducibility, we provide comprehensive hyperparameters for all three training stages, including optimization settings, learning rate schedules, parallelization strategies, and GRPO-specific parameters.

ReBind-Instruct. We build ReBind-Instruct upon Qwen3-Omni-30B-A3B-Instruct, a pretrained multimodal model with audio-visual understanding capabilities, and fully fine-tune it in both stages. Stage 1 performs supervised fine-tuning on 40K video editing triplets for 3 epochs with a global batch size of 128 and a learning rate of 1.0×10^{-5}. Stage 2 applies GRPO to 2K samples with 8 candidates per sample, a learning rate of 1.0×10^{-5}, and a KL coefficient of $\beta = 0.04$ on 32 H200 GPUs.

ReBind-Edit. We build ReBind-Edit upon LTX-2.3 (HaCohen et al., 2026), which uses Gemma-3-12B-IT as its text encoder and a 14B-parameter DiT backbone. ReBind-Edit adapts this model through the lightweight conditioning extension described in the main paper, while fully fine-tuning the DiT backbone. We train ReBind-Edit on approximately 100K video editing triplets for 10K steps with a global batch size of 8 and a learning rate of 1.0×10^{-5}. Following the standard LTX-2.3 configuration, all edited videos are generated at 720P with 81 frames. The model is configured with up to eight ordered reference-image slots, while training samples contain one to five reference images. Unused slots are masked during attention to prevent padding from affecting active reference representations.

Table 6 summarizes the complete training configuration. All stages use mixed-precision training and gradient checkpointing.

B.2 Editor Positional Encoding

We retain the backbone's 3D RoPE and spatial patch indexing. Let $f \in \{0, \dots, F-1\}$ denote a video-frame index in the latent sequence and (h, w) the spatial patch indices. We assign positional index tuples as

$$\begin{aligned} p_{src}(f,h,w) = p_{edit}(f,h,w) &= (f,h,w), \\ p_{ref_k}(h,w) &= (t_{\text{ref}},h,w), \qquad k = 1,\dots,K, \end{aligned} \tag{10}$$

where $t_{\text{ref}} \notin \{0, \dots, F-1\}$ is a fixed index shared by all reference images. These tuples are mapped to rotary phases through the backbone's positional encoding and applied to attention queries and keys. Corresponding source and edited frames share temporal indices, while references occupy a separate static temporal position. Concatenating additional references therefore leaves the video-frame indices unchanged. Unused reference slots are masked during attention. The same index as-

Table 6: Training hyperparameters across three stages: Stage 1 (ReBind-Instruct SFT), Stage 2 (ReBind-Instruct GRPO), and Stage 3 (ReBind-Edit).

Hyperparameters	**Stage 1 (Instructor SFT)**	**Stage 2 (Instructor GRPO)**	**Stage 3 (Editor)**
Learning rate	1.0×10^{-5}	1.0×10^{-5}	1.0×10^{-5}
LR scheduler	Cosine	Cosine	Warmup-Constant-Cosine
LR warmup steps	0.02 (fraction)	-	500
Constant steps	-	-	500
Min learning rate	4.0×10^{-6}	-	1.0×10^{-6}
Weight decay	0.0	0.0	-
Optimizer	AdamW	AdamW	AdamW
Global batch size	128	128	8
Micro batch size	4	1	1
Gradient accumulation	-	-	8
Max sequence length	32768	32768	-
MOE aux loss coeff	1.0×10^{-3}	1.0×10^{-3}	-
Parallelization (Stages 1-2 only)			
Context parallel size	1	1	-
Tensor parallel size	4	4	-
Pipeline parallel size	1	1	-
Expert parallel size	8	8	-
GRPO-specific parameters (Stage 2 only)			
Beta	-	0.04	-
Epsilon (low, high)	-	0.005, 0.01	-
Num generations	-	8	-
Temperature	-	1.0	-
Importance sampling	-	sequence	-

signment is used in training and inference. Positional indices are separate from diffusion timesteps, which are zero for clean conditions and σ_t for edited-video tokens.

The separate temporal position distinguishes static reference conditions from video frames (Wei et al., 2026b). The shared t_{ref} identifies the static-reference input type; it does not encode the ordinal reference identifier k.

B.3 Complete GRPO Objective

GRPO (Shao et al., 2024) optimizes a policy through relative comparisons among candidate outputs sampled for the same input. For each training instance $(V_{src}, V_{edit}, \{I_1, \ldots, I_K\}, S_{gt})$, we sample G candidate instructions $\{S_1, \ldots, S_G\}$ from the rollout policy and compute the composite reward $R(S_g)$ for each candidate. The policy is updated using group-normalized advantages and a clipped objective with KL regularization toward the reference policy. We provide the token-level formulation used by `ms-swift` below.

For group rewards $\{R(S_g)\}_{g=1}^{G}$, the normalized advantage is

$$A_g = \begin{cases} \dfrac{R(S_g) - \mu_R}{\sigma_R}, & \sigma_R > 0, \\ 0, & \sigma_R = 0, \end{cases} \tag{11}$$

where $\mu_R = \frac{1}{G}\sum_{g=1}^{G} R(S_g)$ and σ_R is their standard deviation. With sequence-level importance sampling, the probability ratio is

$$\rho_g(\theta) = \exp\left(\frac{1}{T_g}\sum_{t=1}^{T_g} \log \frac{\pi_\theta(s_{g,t} \mid x, s_{g,<t})}{\pi_{\text{old}}(s_{g,t} \mid x, s_{g,<t})}\right), \tag{12}$$

where π_{old} is the rollout policy and T_g is the number of valid completion tokens. We define $\bar{\rho}_g = \text{clip}(\rho_g, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}})$. The per-token KL estimator is

$$D_{g,t} = \exp(\Delta_{g,t}) - \Delta_{g,t} - 1, \tag{13}$$

where $\Delta_{g,t} = \log \pi_{\text{ref}}(s_{g,t} \mid x, s_{g,<t}) - \log \pi_\theta(s_{g,t} \mid x, s_{g,<t})$ and π_{ref} is the fixed SFT policy. The complete objective is

$$\mathcal{L}_{\text{GRPO}} = \frac{1}{G}\sum_{g=1}^{G}\frac{1}{T_g}\sum_{t=1}^{T_g}\left[-\min(\rho_g A_g, \bar{\rho}_g A_g) + \beta D_{g,t}\right]. \tag{14}$$

We use $\epsilon_{\text{low}} = 0.005$, $\epsilon_{\text{high}} = 0.01$, and $\beta = 0.04$.

Zero-variance reward groups. When all candidates in a group receive the same reward, $\sigma_R = 0$ and Equation 11 assigns $A_g = 0$ to every candidate without dividing by σ_R. The reward-dependent policy-gradient term then vanishes for that group, while the KL regularization term remains. The clipping of ρ_g in Equation 14 applies to the importance ratio and does not itself clip reward values.

Raw checklist scores and reported IQ. For a nonempty checklist, the raw expressions for $\mathcal{R}_{pre}$ and $\mathcal{R}_{diff}$ in Section 4.2 can be negative when the total error count exceeds the number of ground-truth keypoints. Group-relative optimization does not require nonnegative rewards. The reported IQ metric in Section C.1.3 explicitly applies a lower bound of zero; this metric-level bound is distinct from the clipping of the GRPO importance ratio.

C Evaluation Protocols

We specify the evaluation benchmarks and metrics (Section C.1), construction of the instruction-generation evaluation set (Section C.2), and baseline and inference protocols (Section C.3). Section C.4 provides the prompt used for automatic video-editing evaluation on UniVBench.

C.1 Evaluation Benchmarks and Metrics

Video editing is evaluated on IntelligentVBench and UniVBench. Instruction generation is assessed using Instruction Quality (IQ), Reference Accuracy (RA), and their average. We define these benchmarks and metrics below; the source–edited video pairs used for instruction evaluation are described in Section C.2.

C.1.1 IntelligentVBench

IntelligentVBench (Pan et al., 2026) is a comprehensive benchmark designed to assess next-level intelligent unified video generation by integrating abstract reasoning and free-form composition. We evaluate our method on the Text-Image-Video-to-Video (TIV2V) subset, which assesses cross-modal compositional editing capabilities. The TIV2V task requires models to extract specific concepts from reference images and integrate them into source videos while preserving temporal consistency and visual fidelity of non-target elements. This subset comprises 210 test cases spanning three operational objectives: inserting an object from a reference image into the source video (71 cases), substituting a specific visual element within the video with one from the reference image (73 cases), and replacing the video's original background with the scene depicted in the reference image (66 cases). We use Gemini-2.5-Pro as the automatic evaluator. The benchmark reports three metrics rated on a 1–5 scale: Instruction Following (IF) measures the model's capacity to seamlessly integrate video and image in strict adherence to instructions, Condition Preserving (CP) evaluates preservation of the source video's unedited regions and verification that newly introduced elements faithfully conform to reference images, and Visual Quality (VQ) assesses aesthetic quality, temporal consistency, and motion naturalness. For N test samples with per-sample scores s_i^{IF}, s_i^{CP}, and s_i^{VQ}, the reported aggregate scores are

$$\text{AVG} = \frac{1}{N}\sum_{i=1}^{N}\frac{s_i^{\text{IF}} + s_i^{\text{CP}} + s_i^{\text{VQ}}}{3}. \tag{15}$$

$$\text{MIN} = \frac{1}{N}\sum_{i=1}^{N}\min\left\{s_i^{\text{IF}}, s_i^{\text{CP}}, s_i^{\text{VQ}}\right\}. \tag{16}$$

Thus, MIN averages the worst dimension of each sample rather than taking the minimum of the three dataset-level mean scores.

C.1.2 UNIVBENCH

UniVBench (Wei et al., 2026a) is the first unified benchmark providing comprehensive evaluation across video understanding, generation, and editing tasks with multi-shot support and copyright-free content. For reference-guided video editing evaluation, UniVBench employs an agentic evaluation system, UniV-Eval, that decomposes videos into shot-level units and performs fine-grained assessment across eight dimensions: Subject, Background, Action, Camera, Color, Lighting, Style, and Relative Position. These categories are further decomposed into 21 specific sub-dimensions that enable diagnostic feedback at the shot level. The evaluation agent performs per-category comparisons between model outputs and input tuples (source video, reference images, text instructions), producing structured weakness checklists that highlight fine-grained deficiencies. Final scores are aggregated along multiple evaluation dimensions, with each dimension scored on a scale from 0 to 10. While UniVBench originally uses Seed-1.6 as the evaluation LLM, we adopt GPT-5.1 for our experiments to ensure more reliable and consistent evaluation quality. UniVBench contains diverse test cases covering both single-reference and multi-reference video editing scenarios.

C.1.3 REBIND INSTRUCTION METRICS

We report Instruction Quality (IQ), Reference Accuracy (RA), and Overall as percentages. The following definitions produce scores in $[0, 1]$. For sample i, let $\mathcal{K}_i$ denote the ground-truth keypoints and let $\mathcal{M}_i$, $\mathcal{I}_i$, and $\mathcal{H}_i$ denote the missing, incorrectly described, and hallucinated keypoints identified by the evaluator. The per-sample instruction quality is

$$\mathrm{IQ}_i = \max\left(0,\, 1 - \frac{|\mathcal{M}_i| + |\mathcal{I}_i| + |\mathcal{H}_i|}{|\mathcal{K}_i|}\right). \tag{17}$$

For reference attribution, we use the same maximum-cardinality one-to-one matching as in Equations 3–5. For sample i, let M_i and N_i denote the numbers of extracted ground-truth and generated triples. Each extracted entry is a separate matching vertex, including repeated generated entries. The binary matrix $W^{(i)}$ uses semantic agreement for entities and descriptions and exact agreement for reference tokens. The number of matched pairs is

$$\begin{aligned} K_i = \max_{X \in \{0,1\}^{M_i \times N_i}} & \sum_{u=1}^{M_i} \sum_{v=1}^{N_i} X_{uv}, \\ \text{subject to} \quad X_{uv} \le w_{uv}^{(i)}, & \qquad \sum_{v=1}^{N_i} X_{uv} \le 1, \qquad \sum_{u=1}^{M_i} X_{uv} \le 1. \end{aligned} \tag{18}$$

These constraints apply to every valid u, v and impose no ordering requirement. Precision and recall use explicit zero-denominator conventions:

$$P_i = \begin{cases} K_i/N_i, & N_i > 0, \\ 0, & N_i = 0, \end{cases} \qquad R_i = \begin{cases} K_i/M_i, & M_i > 0, \\ 0, & M_i = 0. \end{cases} \tag{19}$$

We compute matching F1 directly from the matched-pair count:

$$\mathrm{RA}_i = \begin{cases} \dfrac{2K_i}{M_i + N_i}, & M_i + N_i > 0, \\ 0, & M_i = N_i = 0. \end{cases} \tag{20}$$

The same rule defines the training reward $\mathcal{R}_C$. It returns zero when both collections are nonempty but have no match, or when only one collection is empty. When both are empty, we also assign zero and retain the sample in the macro-average. For $P_i + R_i > 0$, this expression equals $2P_iR_i/(P_i + R_i)$; computing the count-based expression avoids the undefined harmonic mean when $K_i = 0$. For example, one ground-truth triple and three identical matching generated triples yield $K_i = 1$, $P_i = 1/3$, $R_i = 1$, and $\mathrm{RA}_i = 1/2$. Permuting either collection leaves the score unchanged. The

dataset-level scores are macro-averaged over the N samples:

$$\mathrm{IQ} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{IQ}_i, \tag{21}$$

$$\mathrm{RA} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{RA}_i, \tag{22}$$

$$\mathrm{Overall} = \frac{\mathrm{IQ}+\mathrm{RA}}{2}. \tag{23}$$

Table 3 reports IQ, RA, and Overall as percentages.

C.2 Instruction-Generation Evaluation Set

For each of the 1,240 cases in the UniVBench reference-conditioned editing subset, we use Seedance 2.0 (Seedance Team, 2026) and Kling-o3 (Kuaishou Technology, 2024) to generate candidate edited videos from the source video, ordered reference images, and original sparse operation-centric instruction. The candidates are anonymized and screened under predefined criteria covering instruction execution, reference fidelity, source preservation, and visual quality. We exclude 40 cases for which neither model yields a valid realization and select one qualified edited video for each remaining case. Professional annotators then construct and verify ReBind instructions for the resulting 1,200 source–edited video pairs, covering both single- and multi-reference cases. Table 3 evaluates instruction generation on the multi-reference portion of this set. All evaluated MLLMs receive identical source-video, edited-video, and reference-image inputs.

C.3 Baselines and Inference Protocol

To provide context for the comparative evaluation, we present detailed descriptions of all baseline methods used in our experiments.

Instruction-Generation Baselines. We evaluate Gemini-2.5-Pro (Comanici et al., 2025), Gemini-3.1-Pro (Google, 2026), Qwen2.5-Omni-3B and Qwen2.5-Omni-7B (Xu et al., 2025a), Qwen3-Omni-Captioner-30B-A3B and Qwen3-Omni-Instruct-30B-A3B (Xu et al., 2025b), and Gemma4-31B-it (Google DeepMind, 2025). Each model receives the same source video, edited video, and ordered reference images for instruction generation. Table 3 reports results on the multi-reference portion of the annotated evaluation set described in Appendix C.2.

Benchmark-Specific Editing Baselines. On IntelligentVBench, we compare with VACE-Wan2.1, VACE-LTX, VINO, the query and hidden-state variants of UniVideo, and OmniWeaving; HappyHorse 1.0 (HappyHorse, 2026) and Kling-o3 (2026-03) (Kuaishou Technology, 2024) serve as closed-source references. On the single- and multi-reference subsets of UniVBench, we compare with both UniVideo variants, Kiwi-Edit, and OmniWeaving, with Kling-o3 (2026-03) as the closed-source reference. Tables 2 and 1 report the respective comparisons. In the model-comparison tables, bold and underlined values denote the best and second-best open-source results.

UniVideo (Wei et al., 2025) is a unified framework that integrates video understanding, generation, and editing within a single architecture. It adopts a dual-stream design combining a Multimodal Large Language Model (MLLM) for instruction understanding with a Multimodal DiT (MMDiT) for video generation. UniVideo supports two distinct conditioning modes for reference image integration: (1) *query-based* mode extracts reference features through learned query tokens and injects them via cross-attention, enabling flexible composition, and (2) *hidden-state-based* mode directly incorporates the MLLM's hidden states into the generation process, providing richer semantic guidance. These two modes offer different trade-offs between compositional flexibility and semantic fidelity.

Kiwi-Edit (Lin et al., 2026) is a unified video editing framework designed for instruction-reference-following tasks with single reference images. The method synergizes learnable queries and latent visual features for reference semantic guidance, processing reference images through dual pathways: learnable query tokens for flexible composition and latent visual features for precise visual control. Features are fused with the video generation backbone through cross-attention mechanisms. While

originally designed for single-reference scenarios, we evaluate its performance on multi-reference editing tasks to assess its generalization capability.

OmniWeaving (Pan et al., 2026) is a unified video generation model that combines a multimodal large language model with a latent diffusion model for video synthesis. The framework integrates video generation and editing tasks within a single architecture, trained on massive-scale pretraining data encompassing compositional and reasoning-augmented scenarios. The model employs attention-based feature fusion mechanisms to process interleaved text, multi-image, and video inputs. We evaluate its generalization capability on multi-reference video editing tasks.

VACE (Jiang et al., 2025) is an all-in-one framework for video creation and editing built on Diffusion Transformers. VACE unifies multiple video synthesis tasks including reference-to-video generation, video-to-video editing, and masked video-to-video editing through a Video Condition Unit (VCU) that organizes diverse task inputs into a unified interface. The framework employs a Context Adapter structure to inject task-specific concepts via formalized temporal and spatial representations. We evaluate two variants: VACE-Wan2.1 built on Wan2.1 and VACE-LTX built on LTX.

VINO (Chen et al., 2026a) is a unified visual generator that performs image and video generation and editing within a single framework. VINO couples a vision-language model with a Multimodal Diffusion Transformer (MMDiT), where multimodal inputs (text, images, videos) are encoded as interleaved conditioning tokens to guide the diffusion process. This design supports multi-reference conditioning, long-form instruction following, and coherent identity preservation across static and dynamic content without modality-specific architectural components.

Inference Protocol. All video-editing methods start from the same original benchmark instruction and identically ordered source-video and reference-image inputs. For ReBind-Edit, Qwen3-Omni-30B-A3B-Instruct (Xu et al., 2025b) converts the original instruction into ReBind format using the source video and reference images, without access to a target video. This conversion is part of the ReBind inference pipeline. Each baseline follows its official prompt template and preprocessing procedure; we do not impose ReBind-format instructions on baseline methods.

C.4 UniVBench Evaluation Prompt

The following prompt implements the UniVBench video-editing evaluation described in Section C.1.2. We employ GPT-5.1 as the judge model to assess edited videos across eight dimensions: Subject, Background, Action, Camera, Color, Lighting, Style, and Relative Position. The evaluation balances edit success with temporal consistency, recognizing that successful edit execution is the primary goal while maintaining appropriate consistency when the edit permits.

Complete Multi-Reference Video Editing Evaluation Prompt

You are a rigorous evaluator of reference-image-conditioned video editing. You will receive four inputs: (1) an OriginalVideo represented by timestamped frames, (2) ordered ReferenceImages, (3) an EditInstruction, and (4) a GeneratedVideo represented by timestamped frames.
Evaluate the GeneratedVideo against the EditInstruction and visual inputs. For content explicitly modified by the instruction, assess whether the requested content is correctly drawn from the designated reference image. For content not targeted by the instruction, assess whether it is preserved from the OriginalVideo. Do not penalize changes that are necessary to realize the requested edit.
Assign an integer score from 0 to 10 for each of the following eight dimensions:
1. Subject. Evaluate subject identity, appearance, clothing, attributes, quantity, and temporal consistency. Requested subject edits should match the designated reference; unaffected subjects should remain consistent with the OriginalVideo.
2. Background. Evaluate scene layout, environmental elements, and background consistency. Requested background edits should match the designated reference; unaffected regions should preserve the original scene.
3. Action. Evaluate whether subject actions, gestures, expressions, and motion timing satisfy the instruction while remaining temporally coherent. Preserve original actions when no action change is requested.
4. Camera. Evaluate camera motion, viewpoint, framing, shot scale, and transitions. Requested camera changes should be realized; otherwise, camera behavior should remain consistent with the OriginalVideo.

5. Color. Evaluate color palette, tone, saturation, contrast, and cross-frame consistency. Apply reference-specific color changes only when requested.
6. Lighting. Evaluate illumination direction, intensity, shadows, highlights, and temporal stability. Requested lighting changes should match the instruction or designated reference.
7. Style. Evaluate visual style, texture, rendering characteristics, and stylistic consistency across frames. Requested style changes should be applied without degrading content fidelity.
8. Relative Position. Evaluate spatial relationships among subjects, objects, and the camera. Requested additions, replacements, or removals may alter these relationships; otherwise, preserve the original spatial organization.
Use the following scoring scale consistently: 0–2 indicates severe failure, 3–4 major errors, 5–6 partial satisfaction, 7–8 successful editing with minor defects, and 9–10 near-complete satisfaction with strong temporal consistency.
Return exactly one JSON object using this schema:

```
{
  "Subject": {"score": 0, "reason": "..."},
  "Background": {"score": 0, "reason": "..."},
  "Action": {"score": 0, "reason": "..."},
  "Camera": {"score": 0, "reason": "..."},
  "Color": {"score": 0, "reason": "..."},
  "Lighting": {"score": 0, "reason": "..."},
  "Style": {"score": 0, "reason": "..."},
  "Relative Position": {"score": 0, "reason":
"..."}
}
```

Each reason must briefly identify the principal evidence supporting its score. Do not output additional dimensions or an overall score.

D ADDITIONAL RESULTS AND ANALYSES

We provide complementary evidence for ReBind supervision through qualitative instruction-supervision comparisons (Section D.1), training dynamics (Section D.2), and additional single- and multi-reference editing examples (Section D.3).

D.1 EFFECT OF INSTRUCTION SUPERVISION ON EDITING BEHAVIOR

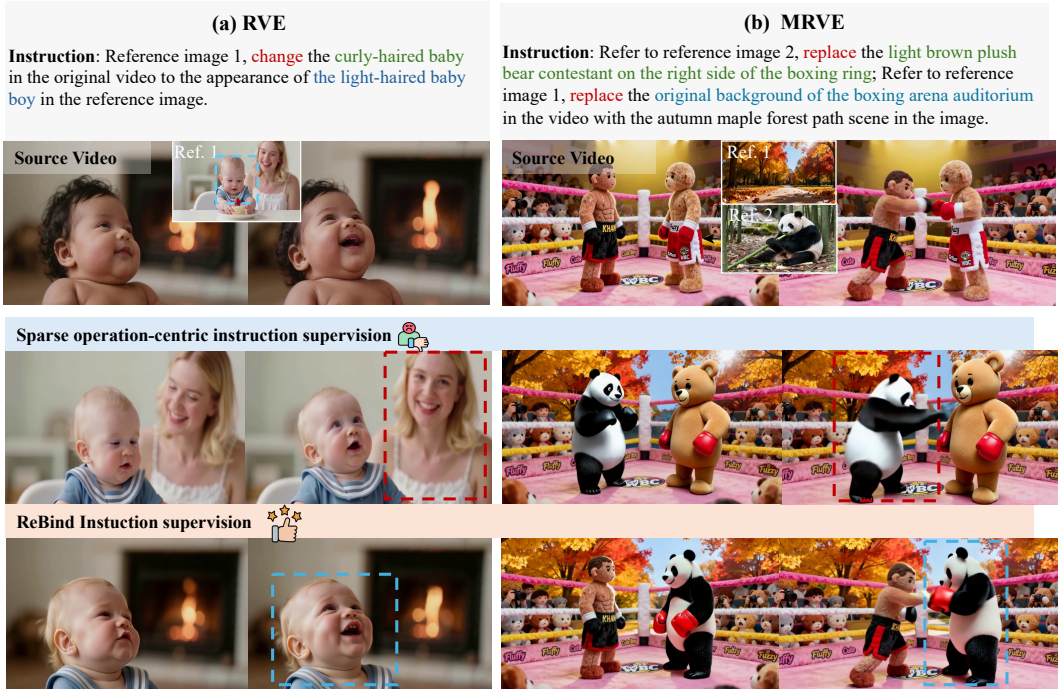

Figure 6: Qualitative comparison of ReBind-Edit trained with different instruction-supervision variants.

Figure 6 complements the controlled comparisons in Table 4 by illustrating how instruction supervision affects editing behavior. In these examples, ReBind-Edit trained with sparse operation-centric instructions exhibits unintended reference transfer and incorrect target binding. ReBind supervision better preserves source content and applies reference attributes to the intended targets.

D.2 TRAINING DYNAMICS

We examine Stage 2 GRPO optimization of ReBind-Instruct and editor training on the instruction-ablation subset. The former tracks the instructor's reward components; the latter documents convergence under the editor ablation budget.

D.2.1 REBIND-INSTRUCT OPTIMIZATION DYNAMICS

Figure 7 tracks Stage 2 GRPO optimization of ReBind-Instruct. The Dual Checklist Reward increases throughout training, while the Referential Consistency Reward remains stable and the Length Reward remains high. The Load Balancing Loss gradually decreases. These trajectories complement the reward-component ablations in Table 5 by documenting the optimization dynamics.

D.2.2 REBIND-EDIT TRAINING DYNAMICS

Figure 8 reports ReBind-Edit training on the 50K randomly sampled cases used for instruction ablations, with ReBind instructions as supervision. The loss decreases rapidly early in training and reaches a stable plateau after approximately 3K steps, with minimal fluctuation through 5K steps.

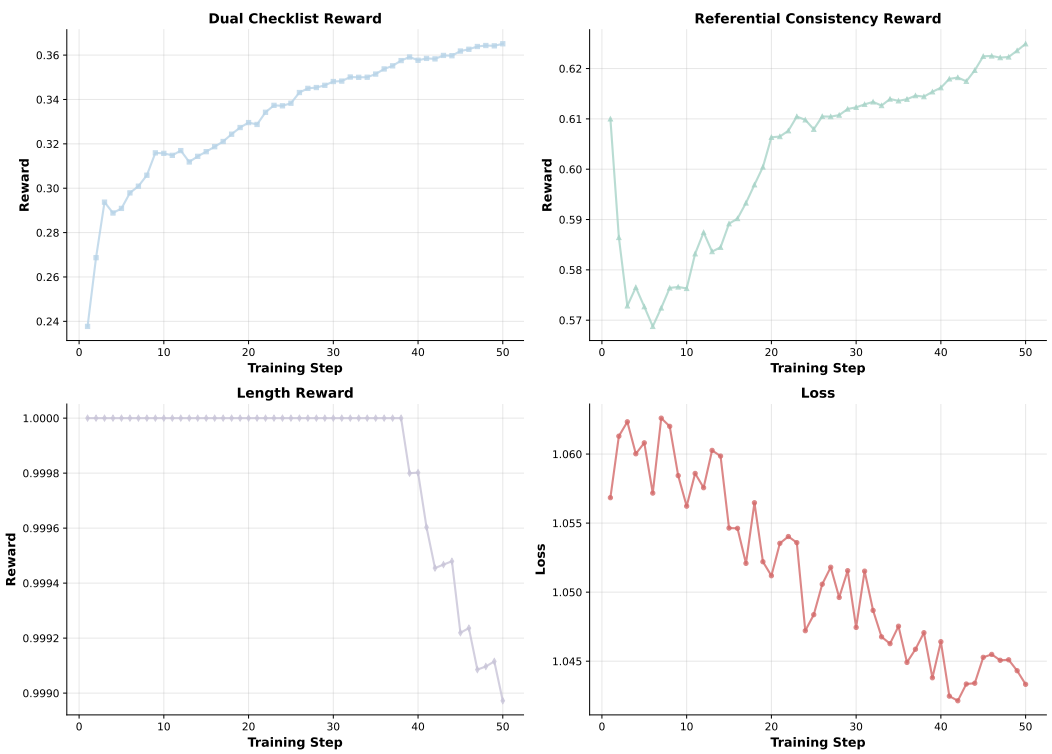

Figure 7: Training dynamics of ReBind-Instruct.

This convergence pattern supports the 5K-step training budget used for the instruction-supervision comparisons in Table 4.

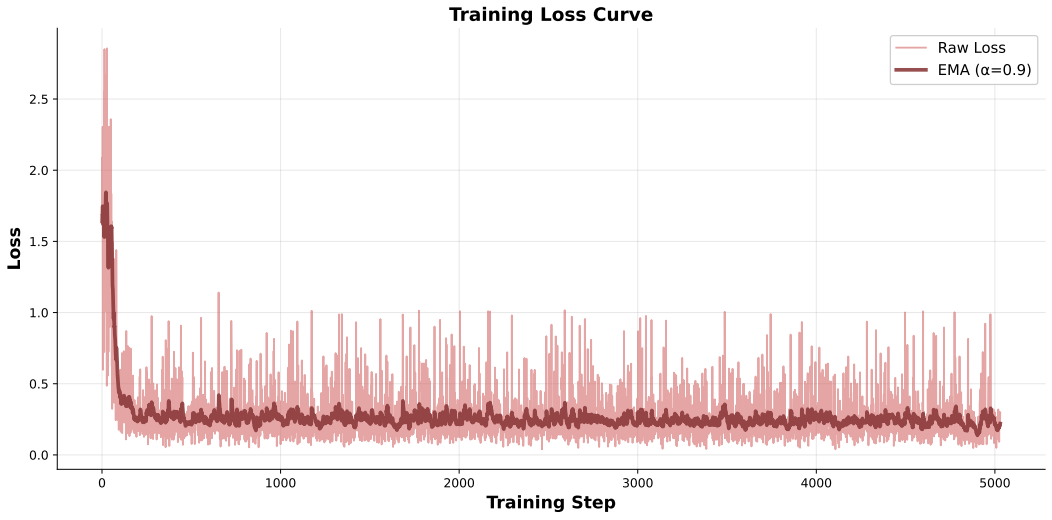

Figure 8: Training loss of ReBind-Edit on the instruction-ablation subset. The loss stabilizes after approximately 3K steps, and all instruction-supervision variants are evaluated at 5K steps.

D.3 Additional Qualitative Results

To provide a more comprehensive view of ReBind's performance across diverse editing scenarios, we present additional qualitative comparisons on both single-reference image-conditioned video editing (RVE) and multi-reference image-conditioned video editing (MRVE) tasks. These examples

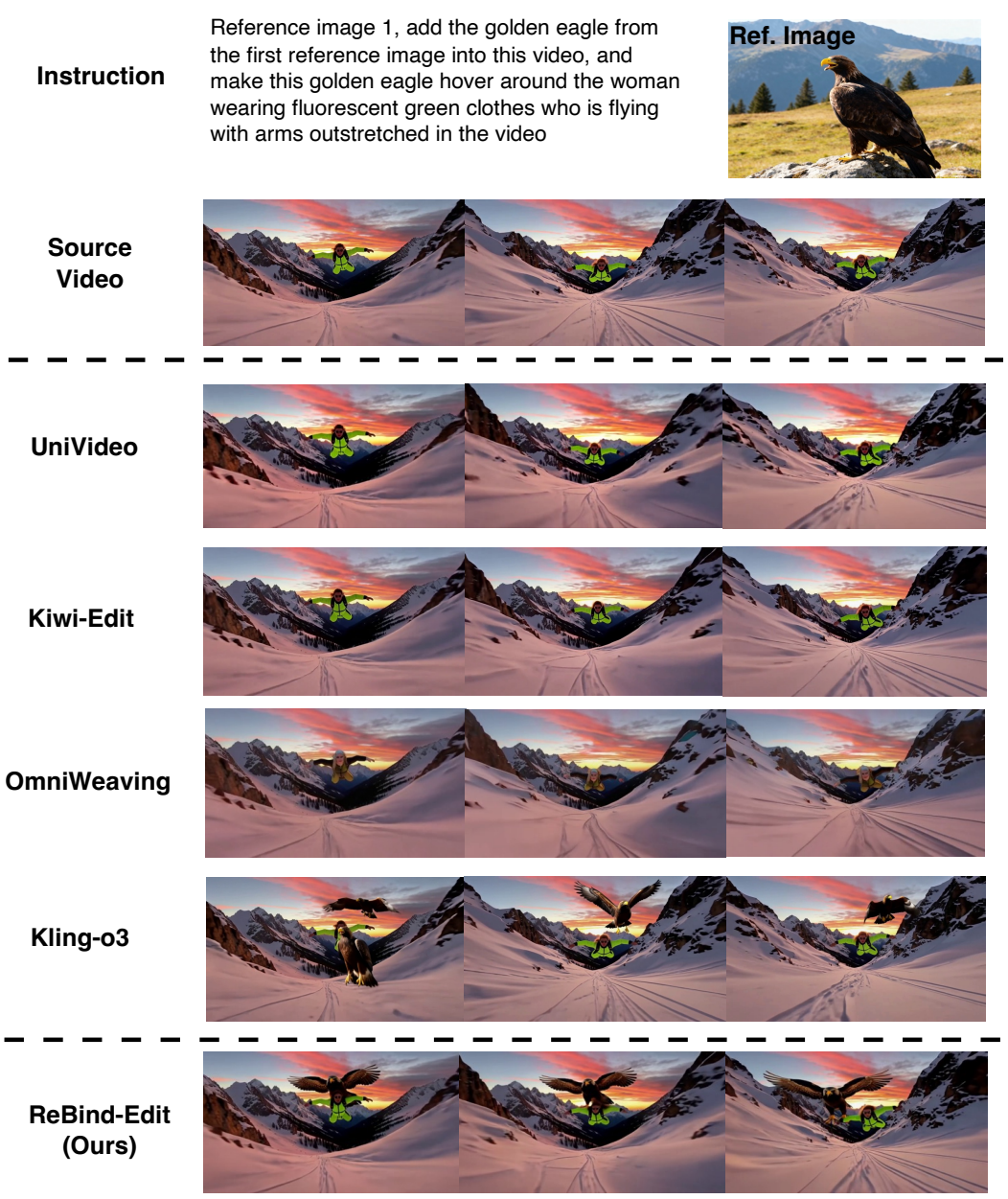

Figure 9: Qualitative comparisons for single-reference image-conditioned video editing on UniVBench.

illustrate precise visual attribute extraction and multi-reference coordination under ReBind supervision.

Single-Reference Editing. Figure 9 presents additional comparisons on UniVBench for single-reference editing tasks, where we compare ReBind-Edit against UniVideo, Kiwi-Edit, OmniWeaving, and the closed-source Kling-o3. In the shown examples, UniVideo and Kiwi-Edit leave some requested operations incomplete, OmniWeaving shows difficulties balancing edit execution and identity consistency, and Kling-o3 exhibits temporal instability in several initial frames. ReBind-Edit illustrates more complete instruction execution, reference-consistent attributes, and coherent temporal behavior while preserving source-video dynamics.

Figure 10 showcases additional results on IntelligentVBench, covering diverse editing operations including local add, background replacement, and local replacement. These examples illustrate

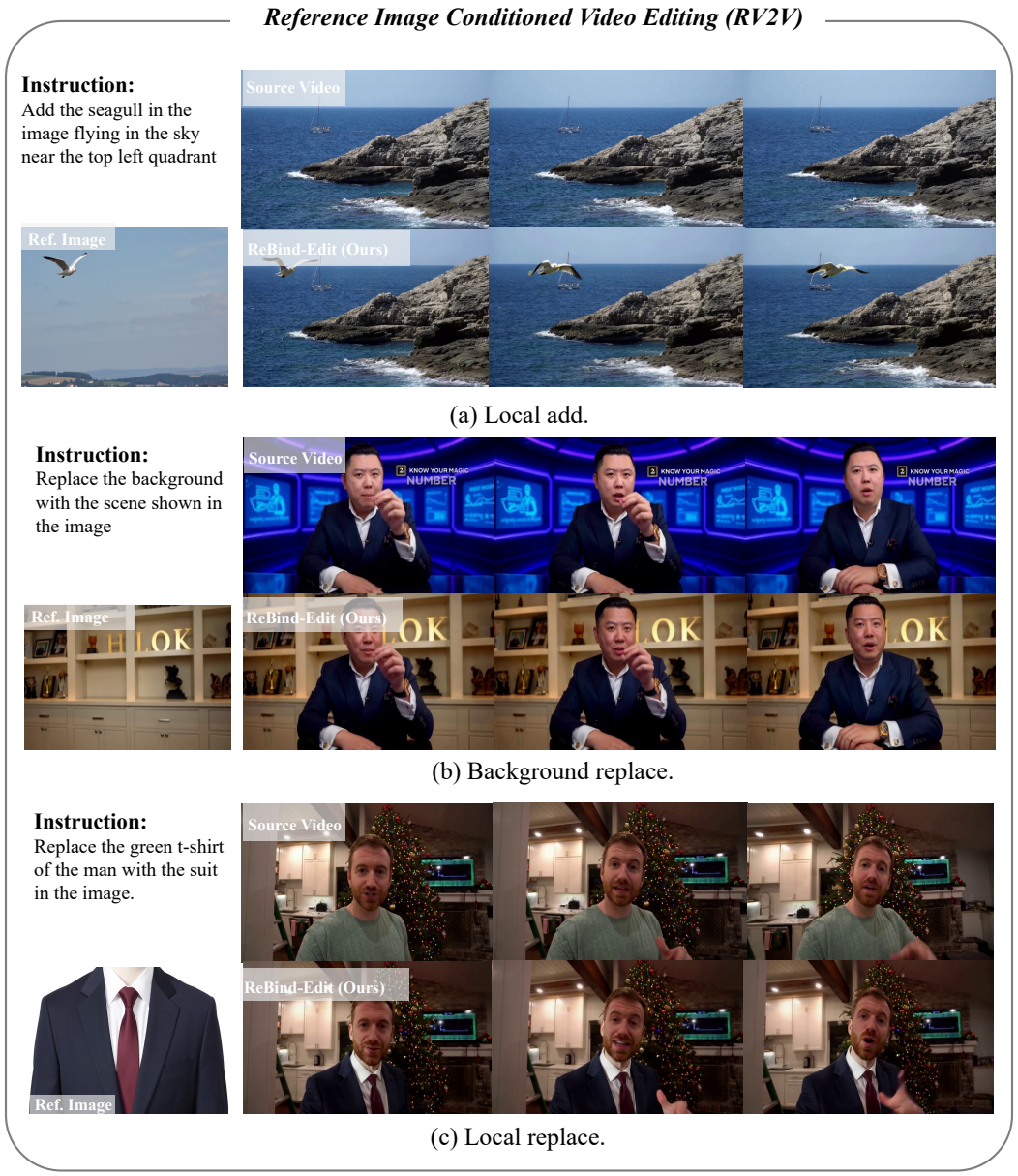

Figure 10: Additional single-reference image-conditioned video editing (RVE) results on IntelligentVBench. Examples span diverse editing types including (a) local add, (b) background replacement, and (c) local replacement.

visually coherent results that follow both the editing instructions and reference visual characteristics across different editing types.

Multi-Reference Editing. Figure 11 presents two challenging three-reference cases. In the first, ReBind-Edit integrates the sofa, carpet, and curtain from distinct references while preserving the source subject and scene structure. The second requires shot-specific identity and accessory edits across a multi-shot video. ReBind-Edit maintains the intended reference assignments across shot transitions, illustrating consistent multi-reference binding in multi-shot editing.



Figure 11: Additional multi-reference image-conditioned video editing (MRVE) results on Uni-VBench.

E LIMITATIONS

This work focuses on visual editing semantics in reference-image-conditioned video editing. ReBind-Instruct incorporates audiovisual context, while ReBind-Edit uses source audio as semantic context and copies it unchanged to the output. Joint audio editing is outside the current scope. The underlying audio-video backbone provides a basis for extending ReBind with audio-specific controls and objectives, together with evaluation of audio fidelity and audiovisual alignment.