BAYESIAN REASONING OVER LATENT EDITING PROGRAMS FOR INSTRUCTION-BASED VIDEO EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Instruction-based video editing requires a model to determine *what* to change, *where* to apply the change across space and time, and *what* to preserve. Existing Diffusion Transformer (DiT) editors learn these decisions only implicitly, providing the same conditioning to every block and supervising attention solely through the generation objective. We propose **ReVEDiT** (**Re**asoning-based **V**ideo **E**diting with **Di**ffusion **T**ransformers), which formulates video editing as hierarchical Bayesian inference over a latent editing program. First, a routing posterior determines the depth at which native textual and visual tokens enter the DiT, so that shallow blocks form the editing intent from MLLM-derived primitives and deeper blocks ground it in the source video. Second, the ideal attention of the edit is treated as a latent variable, for which a reference branch conditioned on the ground-truth edit provides privileged evidence during training; the resulting posterior is distilled into the editing branch at the grounded layers. At inference, the split layer is fixed and the reference branch is discarded, so editing requires a single forward pass. On OpenVE-Bench and EditVerseBench, ReVEDiT achieves the best overall performance among open-source methods under two MLLM judges, with particularly strong results on localized edits.

1 INTRODUCTION

Video has become the dominant medium of visual communication, yet editing it remains a labor-intensive craft that demands expertise in timeline-based software, keyframe manipulation, and frame-level compositing. Recent progress in large language models (LLMs) (Brown et al., 2020; Achiam et al., 2023) and text-conditioned diffusion models (Ho et al., 2020; Rombach et al., 2022) has opened a compelling alternative: instruction-based video editing, in which a user describes a desired modification in natural language and the system returns the edited video automatically. What makes this task hard is not rendering; a modern video diffusion backbone already synthesizes plausible motion and texture. The difficulty is that an instruction radically underdetermines its own realization. Before a single pixel is written, an editor must settle a chain of decisions: which semantic change the instruction actually requests, which spatio-temporal regions carry that change, and what must survive untouched for the entire clip (He et al., 2025). These decisions are tightly coupled: changing too much reduces source fidelity, while handling each frame independently breaks temporal coherence. Because the training data provide only the inputs and final edited video, rather than explicit annotations for these decisions, a Diffusion Transformer (DiT) (Peebles & Xie, 2023) must infer this latent editing program within its forward pass.

Existing methods handle these latent decisions in two broad ways. One line of work delegates them to an external multimodal LLM, which produces an explicit plan, mask, or program for the diffusion model to execute (Yu et al., 2025). Although this decomposition makes the editing process more interpretable, an incorrect plan directly constrains the generated result, with no mechanism for the editor to revise it. Another line learns the decisions implicitly through end-to-end generation. Early methods extend image-editing UNets along the temporal dimension (Brooks et al., 2023; Wu et al., 2023; Qi et al., 2023; Liu et al., 2024), while recent systems improve performance by scaling instruction–video training data and adopting stronger generative backbones (He et al., 2025; Bai et al.; Zi et al., 2025; Zhang et al., 2025). However, neither strategy explicitly models how high-level editing intent should be progressively grounded in local spatio-temporal evidence. In end-to-end approaches, this latent editing process remains underconstrained for two reasons. First, the same conditioning

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.

context is typically supplied to every transformer block (Peebles & Xie, 2023; Esser et al., 2024; Polyak et al., 2024), forcing the model to represent high-level editing intent and fine-grained visual evidence within a single token stream, regardless of depth. Second, flow matching supervises the predicted velocity field rather than the internal attention distributions that connect instruction tokens to spatio-temporal regions (Ho et al., 2020; Lipman et al., 2022; Hertz et al., 2022; Tang et al., 2023). The model must therefore learn where to apply an edit only indirectly from the generation objective, often resulting in coarse token–region correspondences, attribute leakage, object neglect, and imprecise localization (Chefer et al., 2023; Rassin et al., 2023; Agarwal et al., 2023; Patashnik et al., 2023).

We instead infer this latent program inside the backbone. The source video and instruction, which are available during both training and inference, describe *what* change is requested and guide how conditioning should be introduced across transformer depths. However, they do not explicitly reveal *where* the change should be applied: the same source–instruction pair can admit several plausible spatio-temporal realizations. The asymmetry dictates the treatment: latents identifiable at deployment time should be amortized from the source video and instruction pair, whereas those identifiable only under privileged evidence should be inferred with it during training and then distilled into the posterior that remains at test time. We instantiate this principle in **ReVEDiT** (**Re**asoning-based **V**ideo **E**diting with **Di**ffusion **T**ransformers), which casts instruction-based video editing as hierarchical Bayesian inference over the latent editing program. Bayesian granularity routing (§4.1) infers one global split layer from the training corpus, dividing the DiT into two stages: shallow blocks use MLLM-derived editing primitives to establish intent, while deeper blocks additionally receive native textual and visual tokens for local grounding. *Bayesian Latent Attention Reasoning* (§4.2) covers the privileged half: a latent ideal attention trace is observed through a shared-parameter reference branch conditioned on the ground-truth edit, fused with the editing branch by precision weighting, and distilled in attention-score space at the layers routing marks as grounded. Both are training-time only, so the construction adds a single divergence term to flow matching and inference remains one routed forward pass at no extra cost.

In summary, this paper makes the following contributions:

- We formulate instruction-based video editing as hierarchical Bayesian inference over a latent program that separates *what* to edit from *where* to apply the edit, with a single latent routing boundary controlling when native textual and visual evidence enters the backbone.
- We propose ReVEDiT with two coupled components. *Amortized Bayesian Granularity Routing* learns an input-adaptive schedule that progressively introduces textual and visual evidence across transformer depths. *Bayesian Latent Attention Reasoning* uses a shared-parameter reference branch during training to estimate the desired attention pattern and distills it into the editing branch at the layers selected by the routing schedule.
- We validate ReVEDiT on public benchmarks under a dual-judge MLLM evaluation protocol, where it attains the best overall performance among open-source baselines with the largest gains on localized and fidelity-sensitive edits.

2 Related Work

2.1 Instruction-based Video Editing

Instruction-based editing has emerged as a natural interface for generative models, allowing users to modify visual content through free-form natural-language commands rather than carefully engineered source–target prompt pairs. Pioneered in the image domain by InstructPix2Pix Brooks et al. (2023) and its successors Shagidanov et al. (2024); Chen et al. (2025), this paradigm has been progressively transferred to video. Early attempts such as InsV2V Cheng et al. (2023) and InstructVid2Vid Qin et al. (2024) inflate image-level instruction-tuned backbones along the temporal axis, but inherit their limited motion fidelity and suffer from flickering artifacts. More recent efforts push scale along both data and architecture: EffiVED Zhang et al. (2024) constructs large instruction-video triplet corpora through automated pipelines, while Señorita-2M Zi et al. (2025) and related million-scale benchmarks expose the task to diverse real-world edits. Despite this progress, these systems are predominantly built upon UNet-based video diffusion priors Tu et al. (2025); Zhu et al. (2025) and remain agnostic to the structural properties of modern Diffusion Transformers (DiTs) Peebles & Xie (2023), leaving

open the question of how to faithfully ground a textual instruction in the token-level computation of a DiT backbone. More recently, DiT-based methods have begun to emerge Bai et al.; Team (2025); Tan et al. (2025); He et al. (2025); Lin et al. (2026), for instance, Ditto Bai et al. leverages edited keyframes and depth maps, and Kiwi-Edit Lin et al. (2026) and OpenVE-Edit He et al. (2025) integrate vision-language models to enhance instruction following. Despite this progress, all of these methods treat the DiT backbone as a conditioned generator, leaving two structural bottlenecks unaddressed: undifferentiated token conditioning across depths and unsupervised attention geometry, precisely the gaps our method targets.

2.2 Attention and Feature Alignment in Diffusion Models

Attention maps and intermediate features inside diffusion backbones have been repeatedly identified as the locus where semantic correspondence between prompt tokens and visual regions emerges, and aligning or manipulating them has become a principled lever for controllable synthesis. Prompt-to-Prompt Hertz et al. (2022) first demonstrated that swapping or reweighting cross-attention maps enables localized image edits without retraining, while Plug-and-Play Tumanyan et al. (2023) and MasaCtrl Cao et al. (2023) generalize this insight by transferring self-attention features and queries across generation passes to preserve structural fidelity. In the video regime, FateZero Qi et al. (2023), Video-P2P Liu et al. (2024), and TokenFlow Geyer et al. (2023) propagate attention and feature correspondences along the temporal axis to suppress flickering and enforce inter-frame consistency, whereas more recent DiT-oriented studies Mao et al. (2026); Chen et al. (2024) analyze the emergent attention geometry of transformer-based video priors.

These approaches manipulate attention deterministically, mostly at inference time and on frozen backbones. In contrast, we model the ideal attention of an edit as a latent variable and infer it during training. The editing branch and a reference branch conditioned on the ground-truth edit are treated as noisy observations of this variable, and the resulting posterior is distilled into the editing branch at the grounded layers selected by routing. Since the reference branch is discarded after training, inference requires only a single forward pass.

3 Problem Statement

Instruction-based video editing is a conditional generation problem, i.e. a mapping $f_\theta : \mathcal{V} \times \mathcal{I} \to \hat{\mathcal{V}}$, $\hat{v} = f_\theta(v, \ell)$, that takes a source video $v \in \mathcal{V} = \{v \in \mathbb{R}^{T\times H\times W\times 3}\}$ and an instruction $\ell \in \mathcal{I}$ and returns an edited video of the same length and resolution. Unlike unconstrained generation, $\hat{v}$ must be *jointly* faithful to ℓ and to v: it must realize the semantics of the instruction, leave untouched content intact, and stay coherent across time. Collecting these requirements into an editing criterion $\mathcal{J}(\hat{v}, v, \ell) \in \mathbb{R}_{\geq 0}$ that is minimized when $\hat{v}$ is simultaneously instruction-faithful, content-preserving, and temporally consistent, training solves

$$\min_{\theta} \ \mathbb{E}_{(v,\ell)\sim\mathcal{D}}\Big[\, \mathcal{J}\big(f_\theta(v,\ell),\, v,\, \ell\big)\,\Big]. \tag{1}$$

These constraints are inherently coupled: modifying too much to satisfy ℓ compromises source fidelity, while processing frames independently degrades temporal coherence.

We approach Eq. 1 as hierarchical Bayesian inference over a latent editing program, with three families of latent variables: $E_{1:K} \coloneqq \{E_k\}_{k=1}^{K}$, the *semantic editing primitives* representing what should be changed; $R_{1:L} \coloneqq \{R_l\}_{l=1}^{L}$, the *depth-wise routing states* determining when each conditioning signal enters the diffusion transformer; and $U^\star_{1:L} \coloneqq \{U^\star_l\}_{l=1}^{L}$, the *ideal attention traces* characterizing where the edit is grounded. The reference branch available during training is interpreted as privileged Bayesian evidence about the ideal reasoning trajectory, which yields a unified probabilistic account of semantic, routing, and grounding uncertainty while preserving the original flow-matching objective. The resulting reasoning hierarchy is

$$(v, \ell) \to E_{1:K} \to R_{1:L} \to U^\star_{1:L} \to \hat{v}. \tag{2}$$

A compact factorization of the conditional joint model is

$$\begin{aligned} p(\hat{v}, E_{1:K}, R_{1:L}, U^\star_{1:L} \mid v, \ell) = p_\theta(\hat{v} \mid v, \ell, E_{1:K}, R_{1:L}, U^\star_{1:L}) \cdot p_\eta(U^\star_{1:L} \mid R_{1:L}, H_{1:L}, E_{1:K}) \\ \cdot\, p_\psi(R_{1:L} \mid H_{1:L}, E_{1:K}, v, \ell) \cdot p_\phi(E_{1:K} \mid v, \ell). \end{aligned}$$

Conceptually, E answers *what to edit*, R answers *when and at what depth to expose information*, and $U^\star$ answers *where to attend to realize the edit*. The hierarchy splits along the evidence each level can be inferred from: the editing intent and its depth-wise delivery are identifiable from the pair (v, ℓ) that a deployed editor always observes, whereas the grounding of the edit is only weakly constrained by (v, ℓ) and becomes well determined once the reference target seen during training is admitted as additional evidence.

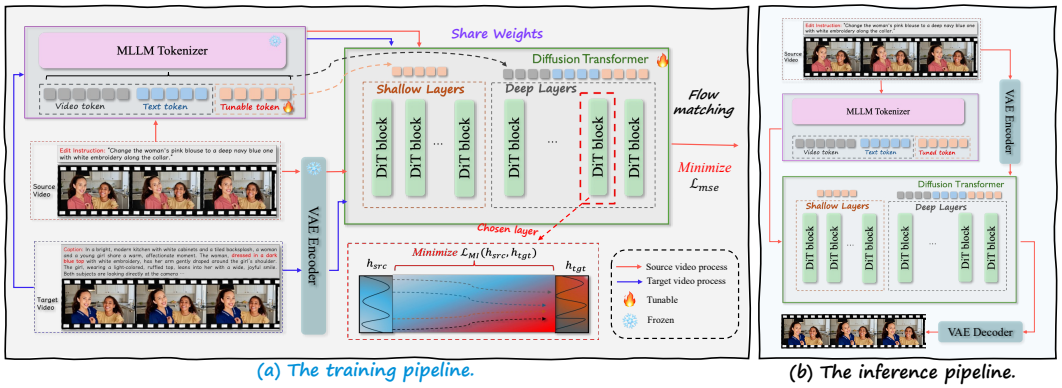


Figure 1: Overview of ReVEDiT. **(a) Training:** An MLLM extracts editing primitives from the source video and instruction. An input-adaptive router progressively introduces textual and visual conditioning across DiT blocks, while a shared-parameter reference branch guides attention at the selected layers. **(b) Inference:** The reference branch is removed, and editing requires only a single routed DiT forward pass.

4 METHODOLOGY

Figure 1 realizes this hierarchy in a diffusion transformer, following the same split in evidence. We first infer the editing intent and the depth at which it is grounded from (v, ℓ) alone, so that the mechanism survives deployment unchanged (§4.1); we then admit the training-time reference as privileged evidence about where the edit should act, and distill it into the posterior that remains available at test time (§4.2). Both levels are trained with a single objective on top of flow matching.

4.1 AMORTIZED BAYESIAN GRANULARITY ROUTING

The signals that serve the coupled demands of Eq. 1 are not equally useful at every depth: the editing primitives E, read out from an MLLM conditioned on (v, ℓ), state the intent abstractly and matter most early, whereas the native tokens X_v and X_ℓ obtained from v and ℓ carry the source appearance and motion against which preservation is measured, and matter most once that intent must be bound to concrete regions. Where this handover should happen is not known a priori, so rather than fixing it as a hand-tuned hyper-parameter we state it as a latent variable and infer it.

Let $H_l \in \mathbb{R}^{N \times d}$ denote the hidden video representation entering DiT block l. We use a compact state summary

$$\bar{H}_l = \mathrm{Pool}(H_l), \tag{3}$$

where pooling can be mean pooling or a lightweight attention-pooling module. Both this section and §4.2 condition on $\bar{H}_l$ rather than on the N spatio-temporal tokens, since both make one decision for the block as a whole. We first write the routing decision at the granularity at which it acts, namely one state per block. For M discrete routing regimes $R_l \in \{1, \dots, M\}$ and

$$q_\psi(R_l \mid \cdot) = \mathrm{Cat}(\pi_l),\ \pi_l = \mathrm{softmax}\big(g_\psi(R_{l-1}, \bar{H}_l, E, v, \ell, l)\big)\,, \tag{4}$$

so that the trajectory posterior factorizes as

$$q_\psi(R_{1:L} \mid v, \ell, E) = q_\psi(R_1 \mid v, \ell, E) \prod_{l=2}^{L} q_\psi(R_l \mid R_{l-1}, \bar{H}_l, E, v, \ell). \tag{5}$$

Conditioning on R_{l-1} turns this reasoning into one coarse-to-fine process rather than L unrelated choices, since what a block should attend to depends on how much of the edit is already settled upstream. For video editing, the early steps decide the content change consistently across frames, such as which object is replaced or which style is applied, and the later steps bind that change to specific regions and refine texture, boundaries, and motion against the source.

The edit requires two ordered regimes: intent formation followed by grounded refinement, so we restrict both the prior and posterior to trajectories with a single transition. Each valid trajectory is then parameterized by one split layer $L_s \in \{1, \dots, L\}$, with $R_l = 1$ for $l \le L_s$ and $R_l = 2$ afterwards. Under this restriction, inferring $R_{1:L}$ is equivalent to inferring L_s. With two regimes, the only free quantity at block l is the probability of switching there, given that no switch has occurred yet,

$$h_l = q_\psi(R_l = 2 \mid R_{l-1} = 1, \bar{H}_l, E, v, \ell). \tag{6}$$

The split layer is s exactly when blocks $1, \dots, s$ stay in the intent regime and block $s+1$ switches, so

$$q_\psi(L_s = s \mid v, \ell, E) = h_{s+1} \prod_{l=1}^{s} (1 - h_l), \tag{7}$$

with $h_1 \coloneqq 0$, so that the first block always forms intent, and $h_{L+1} \coloneqq 1$. The prior $p(L_s)$ is the categorical distribution induced in the same way by the transition prior $p(R_l \mid R_{l-1})$.

At inference, we replace the per-input posterior with a single fixed split layer, the mode of the aggregate posterior over the training data,

$$\hat{L}_s = \arg\max_s \; \mathbb{E}_{(v,\ell)\sim\mathcal{D}}\big[q_\psi(L_s = s \mid v, \ell, E)\big], \tag{8}$$

so deployment requires no evaluation of g_ψ. This trades per-input adaptivity for a fixed schedule, the context supplied to block l is

$$C_l = \begin{cases} E, & l \le L_s, \\ [E; X_\ell; X_v], & l > L_s, \end{cases} \tag{9}$$

so the editing primitives first establish global intent, after which the native tokens provide local evidence for grounding and preservation. The same split also identifies where visual grounding can be supervised (§4.2).

4.2 Bayesian Latent Attention Reasoning

The routing of §4.1 fixes what each depth may look at, but not where the edit should act: which spatio-temporal positions carry the requested change and which are left as they are. This grounding is only weakly constrained by (v, ℓ), and it is where training offers evidence that deployment does not, since the reference target discloses the outcome of a solved edit. We therefore introduce $U_l^\star$, the latent ideal pre-softmax attention logits at layer l, and treat the reference branch as privileged evidence about it. The corresponding ideal attention scores are obtained by row-wise normalization over key positions,

$$A_l^\star = A(U_l^\star), \qquad A(U) \coloneqq \mathrm{softmax}(U), \tag{10}$$

and it is these scores, rather than the logits themselves, that the layer ultimately applies to its values. We use a conditional Gaussian prior

$$p_\eta(U_l^\star \mid R_l, \bar{H}_l, E) = \mathcal{N}(\mu_l^0, \Sigma_l^0), \tag{11}$$

whose parameters $[\mu_l^0, \log \Sigma_l^0]$ are predicted from $(R_l, \bar{H}_l, E)$ by a lightweight head (Appendix C).

The editing branch provides pre-softmax logits U_l^e, and the reference branch, a second pass through the same transformer conditioned on the ground-truth edited video and its caption $(v^\star, \ell^\star)$ and hence available only during training, provides U_l^r. Since the two share parameters and differ only in their conditioning, we interpret both as noisy observations of the same latent ideal trace:

$$U_l^e \mid U_l^\star \sim \mathcal{N}(U_l^\star, (\sigma_l^e)^2 I),\; U_l^r \mid U_l^\star \sim \mathcal{N}(U_l^\star, (\sigma_l^r)^2 I). \tag{12}$$

The uncertainties σ_l^e, σ_l^r are predicted from the branch hidden states rather than chosen by hand, so the posterior over $U_l^\star$ is Gaussian with precision $(\Sigma_l^{\text{post}})^{-1} = (\Sigma_l^0)^{-1} + (\sigma_l^e)^{-2} I + (\sigma_l^r)^{-2} I$ and a mean that weights the prior and the two branches by these precisions (Appendix C). An unreliable

reference trace is therefore downweighted automatically, which matters because a target that departs from the source beyond the requested change should not be copied.

Because the informed reference branch is unavailable at test time, we distinguish a training-time privileged posterior from an inference-time editing posterior:

$$q_\omega^{\text{ref}}(U_l^\star \mid U_l^e, U_l^r, R_l, \bar{H}_l) = \mathcal{N}(\mu_l^{\text{ref}}, \Sigma_l^{\text{ref}}),\ q_\omega^{\text{edit}}(U_l^\star \mid U_l^e, R_l, \bar{H}_l) = \mathcal{N}(\mu_l^{\text{edit}}, \Sigma_l^{\text{edit}}). \tag{13}$$

The quantity that actually drives the forward pass is not the logit field $U_l^\star$ but the attention score $A(U_l^\star)$ of Eq. 10, and the softmax is invariant to a constant shift of each query row. Matching the two posteriors directly in logit space would therefore spend capacity on directions that leave the model's behavior unchanged. We instead transfer the privileged evidence in *score space*: each posterior over $U_l^\star$ induces a posterior predictive over ideal attention maps, with means $\bar{A}_l^{\text{ref}} = \mathbb{E}_{q_\omega^{\text{ref}}}[A(U_l^\star)]$ and $\bar{A}_l^{\text{edit}} = \mathbb{E}_{q_\omega^{\text{edit}}}[A(U_l^\star)]$, estimated with a single reparameterized sample per branch. Writing $\bar{A}_{l,i}$ for the attention distribution of query i, we transfer privileged reasoning information through the row-wise categorical divergence

$$\mathcal{L}_{\text{PD}} = \sum_{l \in \mathcal{S}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{KL}\Big[\bar{A}_{l,i}^{\text{ref}} \,\|\, \bar{A}_{l,i}^{\text{edit}}\Big], \mathcal{S} = \{\, l : R_l = M \,\}, \tag{14}$$

with the reference branch detached so that it acts as a fixed anchor. The signal is thus placed on *where* each query attends, and is insensitive to the arbitrary per-row offset of the underlying logits. Because dense supervision over all of them is memory-prohibitive at video scale, we sample one $l \in \mathcal{S}$ uniformly per iteration, an unbiased estimator of the sum in Eq. 14.

Training Objective Let $\mathcal{Z} = \{E, R_{1:L}, U_{1:L}^\star\}$. Marginalizing $\mathcal{Z}$ out of $p_\theta(\hat{v} \mid v, \ell)$ is intractable, so we bound the conditional evidence under the amortized posterior $q(\mathcal{Z}) = q_\phi(E \mid v, \ell)\, q_\psi(R \mid E, H, v, \ell)\, q_\omega(U^\star \mid R, H)$, which gives

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathcal{Z})}[\mathcal{L}_{\text{FM}}(\hat{v}; v, \ell, \mathcal{Z})] + \beta_E \mathbb{KL}[q_\phi(E \mid v, \ell) \| p(E)] + \beta_R \sum_{l=1}^{L} \mathbb{KL}[q_\psi(R_l \mid \cdot) \| p(R_l \mid R_{l-1})] \\ + \beta_U \sum_{l \in \mathcal{S}} \mathbb{KL}[q_\omega(U_l^\star) \| p_\eta(U_l^\star \mid R_l, H_l, E)] + \lambda_{\text{PD}} \mathcal{L}_{\text{PD}},\end{aligned}$$

where $\mathcal{L}_{\text{FM}}$ is the original flow-matching loss and each latent family contributes one prior-matching term. Training Eq. 15 in full is unnecessary at video scale, where the prior-matching terms mainly slow convergence, so we take the degenerate limit $\beta_E = \beta_R = \beta_U = 0$ and use q_ϕ and q_ψ at their amortized modes, $E = \mu^E$ and $R_l = \arg\max_m \pi_{l,m}$. The routing prior then survives not as a penalty but as the hard coarse-to-fine schedule it induces, with $R_{1:L}$ fixing the conditioning context of Eq. 9 and the layer subset $\mathcal{S}$ of Eq. 14, while the prior-matching term for $U^\star$ is redundant once Eq. 14 pulls the edit posterior toward a reference posterior already formed under the prior of Eq. 11. The objective actually optimized is thus

$$\mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda_{\text{PD}} \mathcal{L}_{\text{PD}}. \tag{15}$$

The entire Bayesian construction therefore adds a single loss term on top of flow matching, and because the reference branch and the privileged posterior are training-time only, inference reduces to one routed forward pass with no additional cost.

5 Experiment

Data Construction We build our training corpus from three public video-editing datasets: OpenVE-3M He et al. (2025), ReCo Zhang et al. (2025), and Ditto-1M Bai et al. (2025a), which together cover appearance modification, object insertion/removal, stylization, and motion-preserving attribute change. To balance editing categories and prevent any single source from dominating, we draw a stratified sample of $10k$ clips per category from each dataset, yielding $120k$ instruction–video–target triplets. Since the released captions are heterogeneous, we re-annotate every target video with Qwen3-VL-8B-Instruct Bai et al. (2025b) to obtain unified, visually grounded descriptions compatible with the reference branch.

Training Setting We instantiate ReVEDiT with Wan2.2-TI2V-5B Wan et al. (2025) (L=34 DiT blocks) as the video backbone and Qwen2.5-VL-3B Bai et al. (2025b) as the MLLM that yields K=512 learnable editing tokens, and train on 81-frame clips at 720×1280. The routing split is L_s=17 and the alignment term uses λ_{align}=0.75 with InfoNCE temperature τ=0.07. The DiT and the MLLM-to-DiT connectors are fully fine-tuned; the MLLM language tower is adapted with rank-64 LoRA Hu et al. (2022) and its vision encoder is frozen. We optimize Eq. 15 with AdamW at learning rate 1×10^{-5} for one epoch in `bf16` on 8 NVIDIA H800 GPUs. We summarize the key protocol below and defer the full configuration to Appendix F.1.

Evaluation We evaluate ReVEDiT on **OpenVE-Bench** and **EditVerseBench** against representative open-source video editors: VACE Jiang et al. (2025), OmniVideo Yang et al. (2026), ICVE Liao et al. (2025), Lucy-Edit Team (2025), DITTO Bai et al., and Kiwi-Edit Lin et al. (2026). Since surrogate metrics correlate weakly with human judgment on open-ended edits, we adopt an *MLLM-as-a-judge* protocol with two complementary judges, Gemini-3.1-Pro Comanici et al. (2025) and Qwen3.5-VL-35B Bai et al. (2025b), each scoring every video on a 1–100 scale along three axes reflecting Eq. equation 1: *Instruction Compliance* (IC), *Consistency Fidelity* (CF), and *Visual Quality* (VQ). We report per-category scores averaged across the two judges; prompt templates and aggregation details are in Appendix F.5. We also conduct the user study in Appendix F.7 to evaluate the user satisfaction with the edited videos.

Table 1: Quantitative comparison on **OpenVE-Bench** and **EditVerseBench**. Scores are on a 1–100 scale and are reported separately for Gemini-3.1-Pro and Qwen3.5-VL-35B. Among open-source methods, **bold** marks the best overall score and underline marks the second-best.

Method	#Params.	Reso.	Overall	Local Change			Local Remove			Local Add			Global Style		
				IC	CF	VQ	IC	CF	VQ	IC	CF	VQ	IC	CF	VQ
EditVerseBench: *Gemini-3.1-Pro*															
VACE Jiang et al. (2025)	14B	1280×720	53.27	63.30	49.37	68.04	25.48	23.98	36.36	49.15	55.87	74.32	57.29	63.34	72.78
Lucy-Edit Team (2025)	5B	1280×704	78.96	84.43	81.68	80.47	65.10	70.66	76.52	71.31	82.92	88.25	76.41	85.15	84.57
ICVE Liao et al. (2025)	13B	384×240	80.19	79.30	76.21	76.07	89.26	86.40	85.54	91.96	87.29	83.34	58.72	64.40	83.81
DITTO Bai et al.	14B	832×480	79.96	83.32	85.84	84.99	61.50	68.75	78.86	82.61	88.92	80.44	87.53	81.67	75.12
Kiwi-Edit Lin et al. (2026)	5B	1280×704	<u>87.51</u>	86.71	83.58	86.10	95.70	92.61	91.32	82.56	89.75	87.20	82.31	85.40	86.93
ReVEDiT (Ours)	5B	1280×704	**88.21**	87.67	85.73	85.22	94.23	88.15	92.33	90.56	89.28	85.18	86.35	87.23	86.64
EditVerseBench: *Qwen3.5-VL-35B*															
VACE Jiang et al. (2025)	14B	1280×720	64.99	52.87	43.60	76.24	62.46	60.77	72.65	58.49	61.21	79.53	71.85	58.63	81.55
Lucy-Edit Team (2025)	5B	1280×704	80.71	64.69	85.82	88.75	70.11	83.34	85.65	74.46	82.58	85.74	82.63	84.93	79.87
ICVE Liao et al. (2025)	13B	384×240	82.64	84.33	87.09	93.50	91.89	89.74	94.51	67.73	65.04	70.03	78.12	82.25	87.48
DITTO Bai et al.	14B	832×480	79.88	56.60	88.35	92.58	74.81	81.34	84.35	86.79	78.36	87.47	80.92	61.33	85.71
Kiwi-Edit Lin et al. (2026)	5B	1280×704	<u>91.56</u>	88.52	92.18	88.03	97.09	96.15	93.37	94.53	93.85	90.14	90.48	89.06	85.27
ReVEDiT (Ours)	5B	1280×704	**92.41**	93.63	92.95	89.11	98.30	97.13	94.57	93.88	91.79	88.55	86.91	90.83	91.29
OpenVE-Bench: *Gemini-3.1-Pro*															
VACE Jiang et al. (2025)	14B	1280×720	16.66	37.15	6.46	18.92	5.76	5.37	9.75	11.34	5.22	14.78	29.79	15.10	40.22
Lucy-Edit Team (2025)	5B	1280×704	64.77	76.38	85.51	90.34	14.41	54.92	63.03	48.34	61.97	73.90	26.50	91.33	90.60
ICVE Liao et al. (2025)	13B	384×240	52.36	61.58	42.85	64.26	38.73	43.05	58.68	38.13	32.54	51.04	60.66	58.93	77.91
DITTO Bai et al.	14B	832×480	36.82	48.29	16.52	57.43	6.05	11.81	36.86	44.25	8.36	49.70	53.79	37.00	71.76
Kiwi-Edit Lin et al. (2026)	5B	1280×704	<u>61.35</u>	81.38	62.23	71.97	66.63	48.27	46.69	47.96	40.46	33.45	70.86	84.19	82.07
ReVEDiT (Ours)	5B	1280×704	**62.48**	69.77	70.35	69.42	71.71	41.59	42.80	49.46	55.58	45.09	67.78	82.76	83.48
OpenVE-Bench: *Qwen3.5-VL-35B*															
VACE Jiang et al. (2025)	14B	1280×720	28.41	32.65	16.77	61.28	8.15	8.15	14.42	14.24	10.81	34.42	36.64	36.43	66.91
Lucy-Edit Team (2025)	5B	1280×704	49.24	68.09	57.14	63.37	7.68	44.69	46.80	37.64	38.30	39.16	16.17	84.38	87.48
ICVE Liao et al. (2025)	13B	384×240	66.56	68.46	76.35	86.11	52.37	63.61	69.05	52.34	59.22	73.69	60.66	58.93	77.91
DITTO Bai et al.	14B	832×480	51.95	51.15	41.63	78.49	11.20	25.78	37.95	45.67	35.70	62.30	74.36	73.64	85.57
Kiwi-Edit Lin et al. (2026)	5B	1280×704	<u>75.89</u>	85.28	85.46	88.05	82.29	63.58	64.44	63.94	63.33	73.84	73.02	86.66	80.79
ReVEDiT (Ours)	5B	1280×704	**77.42**	78.71	86.43	89.92	84.32	67.08	67.93	59.49	71.73	79.67	72.50	87.64	83.64

5.1 Main Results

Quantitative Results As shown in Table 1, ReVEDiT achieves the best overall performance on both OpenVE-Bench and EditVerseBench under both MLLM judges, consistently outperforming all open-source baselines, including the strongest competitor, Kiwi-Edit. The improvements are most evident on localized edits, which demand fine-grained alignment between the instruction and specific spatio-temporal regions: ReVEDiT modifies the intended region more faithfully while better preserving its surroundings. These results reflect the strengths of the proposed Bayesian reasoning. Routing lets the model settle *what* to edit from the editing primitives before dense source tokens are introduced, so that the intent is not overridden by source appearance. Posterior distillation then transfers privileged evidence about *where* the edit should act, sharpening the attention of the grounded layers on the regions that must change. Meanwhile, strong performance on global stylization shows that this fine-grained grounding does not come at the cost of scene-level edits.

Qualitative Results Figure 2 compares ReVEDiT against the three most recent baselines, with the full comparison deferred to Appendix F.3 and Appendix F.4. ReVEDiT places edits precisely while preserving identity, layout, and camera motion, consistent with its input-adaptive routing of editing intent and fine-grained visual evidence across transformer depths. On *Local Remove* and *Global Style*, it further yields clean erasures and temporally stable restylings, indicating that reference-guided attention distillation improves spatial and temporal grounding. These trends mirror Table 1.

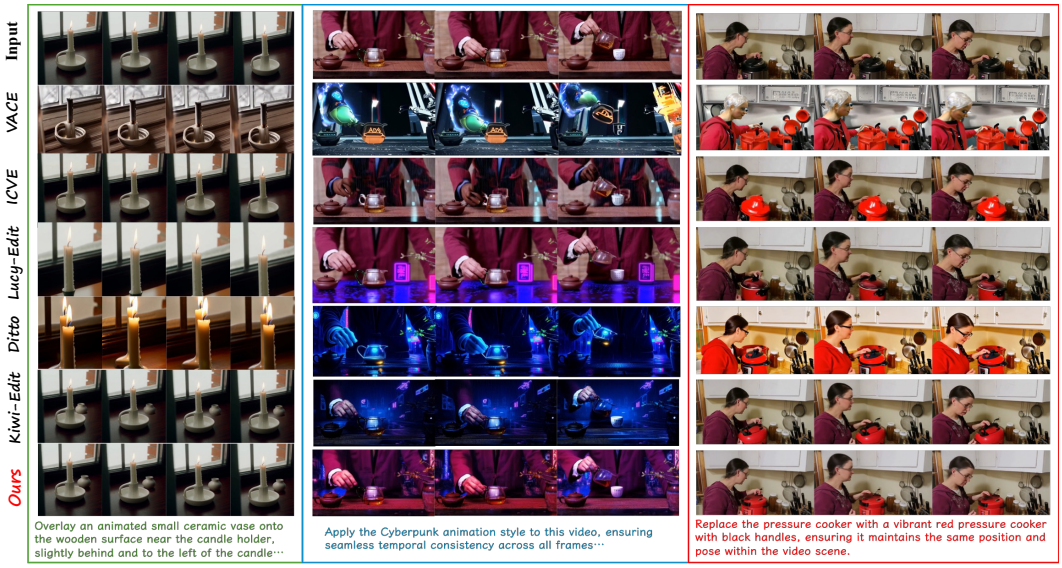


Figure 2: Qualitative comparison across the three edit categories. ReVEDiT localizes the edit precisely, preserves non-edited content, and remains temporally stable, whereas baselines display mislocalization, leakage, residual traces, or over-stylization.

5.2 Ablation and Parameter Study

We ablate the two reasoning components of ReVEDiT: Amortized Bayesian Granularity Routing (§4.1) (w/o granularity routing) and Bayesian Latent Attention Reasoning (§4.2) (w/o latent attention reasoning). All variants share the same training recipe and are evaluated on **OpenVE-Bench** with the Qwen3.5-VL-35B judge. Table 2 reports (a) component ablations, (b) deterministic split layers, which validate the inferred split layer $\hat{L}_s{=}15$, and (c) a sweep over the posterior-distillation weight λ_{PD} in Eq. equation 15.

Table 2: Ablation on **OpenVE-Bench** under the Qwen3.5-VL-35B. Scores are per-category averages over the three evaluation axes; *Overall* further averages across the four categories.

Variant	Overall	Local Change	Local Remove	Local Add	Global Style
ReVEDiT (full, $L_s{=}15$, $\lambda_{\mathrm{PD}}{=}0.75$)	**77.42**	**85.02**	**73.11**	**70.29**	**81.29**
(a) Component ablations					
w/o granularity routing ($L_s{=}L{=}34$)	74.76	84.59	71.02	65.64	77.80
w/o latent attention reasoning ($\lambda_{\mathrm{PD}}{=}0$)	76.06	84.94	73.48	65.74	80.11
w/o both (vanilla DiT editor)	71.71	83.88	56.18	67.06	79.96
(b) Deterministic split layer L_s (with $\lambda_{\mathrm{PD}}{=}0.75$ fixed)					
$L_s{=}8$	74.13	86.48	67.95	63.50	78.61
$L_s{=}15$ (ours)	77.42	85.02	73.11	70.29	81.29
$L_s{=}22$	69.83	80.55	63.87	58.98	76.05
$L_s{=}L{=}34$ (= w/o adaptive routing)	74.76	84.59	71.02	65.64	77.80
(c) Posterior-distillation weight λ_{PD} (with $L_s{=}15$ fixed)					
$\lambda_{\mathrm{PD}}{=}0$ (= w/o attention distillation)	76.06	84.94	73.48	65.74	80.11
$\lambda_{\mathrm{PD}}{=}0.25$	75.09	83.44	71.85	65.65	79.45
$\lambda_{\mathrm{PD}}{=}0.50$	72.75	82.94	64.67	65.27	78.13
$\lambda_{\mathrm{PD}}{=}0.75$ (ours)	77.42	85.02	73.11	70.29	81.29
$\lambda_{\mathrm{PD}}{=}1.00$	73.37	81.76	71.18	61.61	78.93



Figure 3: Qualitative examples of the component ablations reported in Table 2.

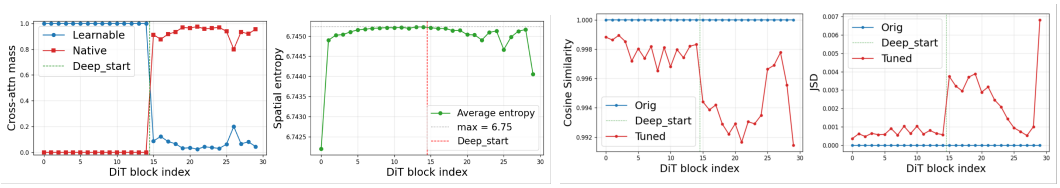
Ablation Analysis Table 2 and Figure 3 expose distinct failure modes of the two mechanisms. Removing adaptive routing degrades the categories that require a clear global editing intent, notably *Local Add* and *Global Style*, and the resulting edits often drift from the target region or leak style into preserved content. Removing reference-guided attention distillation yields a smaller overall drop, but

the degradation is concentrated in categories that depend on fine-grained positional correspondence, consistent with the role of the posterior-distillation objective in Eq. 14. Removing both mechanisms leads to a substantially larger failure, with *Local Remove* dropping by nearly seventeen points overall; Figure 3 further shows compounded residual traces and structural corruption.

Parameter Analysis The parameter sweeps support the chosen setting as the best operating point. Performance peaks at an intermediate routing transition depth L_s, while both smaller and larger values reduce effectiveness, consistent with the need to separate early global-intent formation from later local refinement. Performance is also non-monotonic in λ_{PD}: moderate posterior distillation improves editing quality, whereas excessive weighting biases the model toward the reference attention pattern and weakens instruction following. A detailed per-axis breakdown and additional sweep results are provided in Appendix F.2.

5.3 Method Analysis

We further examine whether the internal behavior of the DiT matches the reasoning process introduced in §4. To analyze Amortized Bayesian Granularity Routing, we measure how cross-attention mass is distributed between the editing primitives E and the native visual/textual tokens across transformer depth, together with the spatial entropy of the resulting attention maps. To analyze Bayesian Latent Attention Reasoning, we compare the attention traces of the editing and reference branches using cosine similarity and JS divergence, and contrast them with the original base model.



(a) Effect of Amortized Bayesian Granularity Routing. (b) Effect of Bayesian Latent Attention Reasoning.

Figure 4: Mechanistic analysis of ReVEDiT. (a) Attention mass on learnable tokens and native visual/textual tokens across DiT depth, with spatial attention entropy. (b) Cosine similarity and JS divergence between the editing and reference attention traces, compared with the base model.

Figure 4(a) characterizes how conditioning changes across depth under Eq. equation 9. Early blocks assign more attention to the MLLM-derived editing primitives E, whereas later blocks shift attention toward native visual and textual tokens and exhibit lower spatial entropy. This pattern is consistent with the intended coarse-to-fine progression from high-level intent formation to localized visual grounding. Figure 4(b) evaluates the effect of Bayesian Latent Attention Reasoning. Relative to the base model, the trained model exhibits greater agreement between the editing and reference attention traces while maintaining non-zero divergence. This suggests that the editing branch transfers useful correspondence structure from the reference without simply reproducing its attention maps. Together with the ablations in Table 2, these analyses support the complementary roles of the two components.

6 Conclusion

We presented ReVEDiT, which formulates instruction-based video editing as hierarchical Bayesian inference over a latent editing program that specifies *what* to edit, *when* to introduce source evidence, and *where* to ground the change. Bayesian Granularity Routing infers the depth at which native textual and visual tokens enter the DiT, so that shallow blocks form the editing intent and deeper blocks ground it in the source video. Bayesian Latent Attention Reasoning treats the ideal attention of an edit as a latent variable, uses a reference branch conditioned on the ground-truth edit as privileged evidence during training, and distills the resulting posterior into the editing branch. After training, routing uses a fixed split layer and the reference branch is discarded, so inference requires only a single routed forward pass. On OpenVE-Bench and EditVerseBench, ReVEDiT achieves the best overall performance among open-source methods under both MLLM judges, and a user study further shows that its edits are consistently preferred by human raters. More broadly, our results suggest that making the implicit reasoning of generative editors explicit, and inferring each decision from the evidence that identifies it, is a promising direction for controllable video generation.

7 AI USE STATEMENT

Generative AI tools were used to assist with language editing, including improving grammar, clarity, conciseness, and the organization of the manuscript. All research ideas, methodological design, experiments, results, and conclusions were developed and verified by the authors. The authors reviewed and revised all AI-assisted text and take full responsibility for the content of the paper.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Aishwarya Agarwal, Srikrishna Karanam, KJ Joseph, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. A-star: Test-time attention segregation and retention for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2283–2293, 2023.

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

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.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22560–22570, 2023.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Hila Chefer, Shiran Zada, Roni Paiss, Ariel Ephrat, Omer Tov, Michael Rubinstein, Lior Wolf, Tali Dekel, Tomer Michaeli, and Inbar Mosseri. Still-moving: Customized video generation without customized video data. *ACM Transactions on Graphics (TOG)*, 43(6):1–11, 2024.

Sherry X Chen, Misha Sra, and Pradeep Sen. Instruct-clip: Improving instruction-guided image editing with automated data refinement using contrastive learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 28513–28522, 2025.

Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6441–6451, 2024.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *arXiv preprint arXiv:2311.00213*, 2023.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023.

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.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Yuming Jiang, Tianxing Wu, Shuai Yang, Chenyang Si, Dahua Lin, Yu Qiao, Chen Change Loy, and Ziwei Liu. Videobooth: Diffusion-based video generation with image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6689–6700, 2024.

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.

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024.

Xinyao Liao, Xianfang Zeng, Ziye Song, Zhoujie Fu, Gang Yu, and Guosheng Lin. In-context learning with unpaired clips for instruction-based video editing. *arXiv preprint arXiv:2510.14648*, 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.

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

Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14951–14961, 2025.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8599–8608, 2024.

Yuxin Mao, Zhen Qin, Xuyang Shen, Jinxing Zhou, Jing Zhang, Yiran Zhong, and Yuchao Dai. Ldt: Efficient scalable video generation using linear diffusion transformer. *IEEE Transactions on Multimedia*, 2026.

Wenqi Ouyang, Yi Dong, Lei Yang, Jianlou Si, and Xingang Pan. I2vedit: First-frame-guided video editing via image-to-video diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23051–23061, 2023.

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.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 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.

Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. Linguistic binding in diffusion models: Enhancing attribute correspondence through attention map alignment. *Advances in Neural Information Processing Systems*, 36:3536–3559, 2023.

Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, and Wenhu Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.

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

Artur Shagidanov, Hayk Poghosyan, Xinyu Gong, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Grounded-instruct-pix2pix: Improving instruction based image editing with automatic target grounding. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6585–6589. IEEE, 2024.

Sirnam Swetha, Jinyu Yang, Tal Neiman, Mamshad Nayeem Rizve, Son Tran, Benjamin Yao, Trishul Chilimbi, and Mubarak Shah. X-former: Unifying contrastive and reconstruction learning for mllms. *arXiv preprint arXiv:2407.13851*, 2024.

Zhiyu Tan, Hao Yang, Luozheng Qin, Jia Gong, Mengping Yang, and Hao Li. Omni-video: Democratizing unified video understanding and generation. *arXiv preprint arXiv:2507.06119*, 2025.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Türe. What the daam: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5644–5659, 2023.

DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025.

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.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1921–1930, 2023.

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

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7623–7633, 2023.

Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision*, 133(3):1175–1194, 2025.

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. *arXiv preprint arXiv:2602.08820*, 2026.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Shoubin Yu, Difan Liu, Ziqiao Ma, Yicong Hong, Yang Zhou, Hao Tan, Joyce Chai, and Mohit Bansal. Veggie: Instructional editing and reasoning video concepts with grounded generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15147–15158, 2025.

Zhenghao Zhang, Zuozhuo Dai, Long Qin, and Weizhi Wang. Effived: Efficient video editing via text-instruction diffusion models. *arXiv preprint arXiv:2403.11568*, 2024.

Zhongwei Zhang, Fuchen Long, Wei Li, Zhaofan Qiu, Wu Liu, Ting Yao, and Tao Mei. Region-constraint in-context generation for instructional video editing. *arXiv preprint arXiv:2512.17650*, 2025.

Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*, pp. 273–290. Springer, 2024.

Yixuan Zhu, Haolin Wang, Shilin Ma, Wenliang Zhao, Yansong Tang, Lei Chen, and Jie Zhou. Fade: Frequency-aware diffusion model factorization for video editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28426–28435, 2025.

Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Se\~norita-2m: A high-quality instruction-based dataset for general video editing by video specialists. *arXiv preprint arXiv:2502.06734*, 2025.

The video samples can refer to `https://anonymous.4open.science/r/RVEDiT-F8A4`.

A LIMITATIONS

While ReVEDiT improves instruction-based video editing with DiT backbones, several limitations remain. The method inherits the capabilities and biases of the underlying video backbone and MLLM; difficult instructions involving rare objects, ambiguous spatial relations, or long-range temporal dependencies may still lead to incorrect or incomplete edits. Future work could improve robustness by incorporating stronger temporal reasoning, more diverse editing data, and human-aligned evaluation protocols for subtle semantic and temporal failures.

B RELATED WORK

B.1 REFERENCE-GUIDED VIDEO GENERATION AND EDITING

A complementary line of work conditions generation on an auxiliary visual reference, supplying appearance or motion cues that are difficult to specify in language alone. In the image domain, IP-Adapter Ye et al. (2023) and subsequent subject-driven variants Xiao et al. (2025) inject reference features through decoupled cross-attention, whereas Q-Former, style resamplers Swetha et al. (2024) compress reference signals into a compact set of learnable queries for downstream conditioning. Extending these ideas to video, VideoBooth Jiang et al. (2024), Still-Moving Chefer et al. (2024), and MotionDirector Zhao et al. (2024) disentangle subject appearance from motion dynamics, while ConsistI2V Ren et al. (2024) and Phantom Liu et al. (2025) exploit a reference frame or clip to enforce temporal coherence. Training-free alternatives such as AnyV2V Ku et al. (2024) and related first-frame-anchored pipelines Ouyang et al. (2024) propagate edits from a single edited keyframe across the clip via feature injection. These methods convincingly demonstrate that a reference branch can stabilize identity and layout, yet they typically treat the reference as a *global* conditioning signal and do not explicitly align its spatio-temporal attention footprint with that of the edited stream—an alignment we show to be crucial for instruction faithfulness under DiT backbones.

C DETAILS OF BAYESIAN LATENT ATTENTION REASONING

This section collects the parameterizations and closed forms omitted from §4.2.

Amortized prior and observation uncertainties The parameters of the conditional prior in Eq. 11 are predicted by a lightweight head,

$$[\mu_l^0, \log \Sigma_l^0] = g_\eta(R_l, \bar{H}_l, E), \tag{16}$$

while the two branches of §4.2 read their pre-softmax logits off their own queries and keys,

$$U_l^e = \frac{Q_l^e (K_l^e)^\top}{\sqrt{d_k}}, \qquad U_l^r = \frac{Q_l^r (K_l^r)^\top}{\sqrt{d_k}}. \tag{17}$$

The scalar observation uncertainties of Eq. 12 are predicted from the branch hidden states,

$$\sigma_l^e = \epsilon + \mathrm{softplus}(g_e(\bar{H}_l^e)), \tag{18}$$

$$\sigma_l^r = \epsilon + \mathrm{softplus}(g_r(\bar{H}_l^r)), \tag{19}$$

with $\epsilon > 0$ a small constant ensuring positivity.

Precision-weighted posterior Combining the Gaussian prior of Eq. 11 with the two Gaussian observations of Eq. 12 gives a Gaussian posterior over $U_l^\star$ with precision

$$(\Sigma_l^{\text{post}})^{-1} = (\Sigma_l^0)^{-1} + \frac{1}{(\sigma_l^e)^2} I + \frac{1}{(\sigma_l^r)^2} I, \tag{20}$$

and mean

$$\mu_l^{\text{post}} = \Sigma_l^{\text{post}} \left[(\Sigma_l^0)^{-1} \mu_l^0 + \frac{U_l^e}{(\sigma_l^e)^2} + \frac{U_l^r}{(\sigma_l^r)^2} \right]. \tag{21}$$

The posteriors q_ω^{ref} and q_ω^{edit} of §4.2 are the two instances of this update obtained with and without the reference observation U_l^r, and supply the means entering the variational bound of Eq. 15.

D ALGORITHM

We summarize the training and inference procedures of ReVEDiT in Algorithm 1 and Algorithm 2. Training optimizes the objective of Eq. 15, which combines the flow-matching loss $\mathcal{L}_{\mathrm{FM}}$, the routing KL term of Bayesian Granularity Routing (§4.1), and the posterior-distillation loss $\mathcal{L}_{\mathrm{PD}}$ of Bayesian Latent Attention Reasoning (Eq. 14). At inference, routing uses the fixed split layer $\hat{L}_s$ of Eq. 8 and the reference branch is discarded, so editing requires a single routed DiT forward pass with no additional cost relative to a vanilla diffusion transformer.

Algorithm 1 Training ReVEDiT with Bayesian Granularity Routing and Bayesian Latent Attention Reasoning.

Require: Dataset $\mathcal{D} = \{(v, \ell, v^\star, \ell^\star)\}$; DiT f_θ with L blocks; MLLM g_ϕ; visual / textual tokenizers $\mathcal{T}_v, \mathcal{T}_\ell$; routing network g_ψ and prior $p(L_s)$; attention prior and posterior heads (η, ω); weights $\beta_R, \lambda_{\mathrm{PD}}$; optimizer Opt.

1: **while** not converged **do**
2: Sample a minibatch $(v, \ell, v^\star, \ell^\star) \sim \mathcal{D}$, flow-matching time $t \sim \mathcal{U}[0, 1]$, and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.
// Tokenize both branches.
3: $X_v \leftarrow \mathcal{T}_v(v), \quad X_\ell \leftarrow \mathcal{T}_\ell(\ell); \quad X_v^\star \leftarrow \mathcal{T}_v(v^\star), \quad X_\ell^\star \leftarrow \mathcal{T}_\ell(\ell^\star)$.
// Semantic editing primitives from the MLLM.
4: $E \leftarrow g_\phi(v, \ell); \quad E^\star \leftarrow g_\phi(v^\star, \ell^\star)$.
// Bayesian Granularity Routing (§4.1).
5: g_ψ decides, block by block, whether the native tokens are introduced, which determines the split layer L_s of each example and its posterior $q_\psi(L_s \mid v, \ell, E)$.
6: Build block contexts C_l by Eq. 9 (analogously $C_l^\star$ for the reference branch).
// Two forward passes through the shared DiT f_θ.
7: Editing branch: predict velocity $\hat{u}_\theta(v_t, t \mid C)$ and record pre-softmax attention logits $\{U_l^e\}$.
8: Reference branch: forward with $C^\star$ on noisy $v_t^\star$ and record $\{U_l^r\}$.
9: $\mathcal{L}_{\mathrm{FM}} \leftarrow \|\hat{u}_\theta(v_t, t \mid C) - u^\star(v, v^\star, t)\|_2^2$.
// Bayesian Latent Attention Reasoning (§4.2).
10: Sample one grounded layer $l \sim \mathcal{U}(\mathcal{S})$, with $\mathcal{S} = \{l : l > L_s\}$.
11: Form $q_\omega^{\mathrm{ref}}(U_l^\star)$ from (U_l^e, U_l^r) and $q_\omega^{\mathrm{edit}}(U_l^\star)$ from U_l^e via Eq. 21.
12: Compute $\bar{A}_l^{\mathrm{ref}}$ and $\bar{A}_l^{\mathrm{edit}}$ with one reparameterized sample each, and $\mathcal{L}_{\mathrm{PD}}$ by Eq. 14 with the reference branch detached.
13: $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{FM}} + \mathbb{KL}[q_\psi(L_s \mid v, \ell, E) \,\|\, p(L_s)] + \lambda_{\mathrm{PD}}\, \mathcal{L}_{\mathrm{PD}}$ (Eq. 15)
14: Update $(\theta, \phi, \psi, \eta, \omega)$ with Opt.
15: **end while**
16: Compute the fixed split layer $\hat{L}_s$ by Eq. 8.
17: **return** trained parameters and $\hat{L}_s$.

Algorithm 2 Inference with ReVEDiT: fixed split layer, no reference branch.

Require: Source video v; instruction ℓ; trained DiT f_θ and MLLM g_ϕ; fixed split layer $\hat{L}_s$; number of sampling steps T; scheduler $\{t_k\}_{k=1}^T$.

1: $X_v \leftarrow \mathcal{T}_v(v), \quad X_\ell \leftarrow \mathcal{T}_\ell(\ell), \quad E \leftarrow g_\phi(v, \ell)$.
2: Build block contexts C_l by Eq. 9 with $L_s = \hat{L}_s$.
3: Initialize latent $z_T \sim \mathcal{N}(0, \mathbf{I})$.
4: **for** $k = T, T-1, \ldots, 1$ **do**
5: $z_{k-1} \leftarrow \mathrm{Step}(z_k,\ \hat{u}_\theta(z_k, t_k \mid C),\ t_k)$.
6: **end for**
7: **return** decoded edited video $\hat{v} = \mathrm{Decode}(z_0)$.

E VISUALIZATION

Figures 5–8 present additional qualitative results of ReVEDiT on the four editing categories of OpenVE-Bench. For localized edits (Local Add, Local Remove, and Local Change), ReVEDiT

modifies only the region specified by the instruction and keeps the remaining content and motion unchanged across frames, consistent with routing first settling what to edit and posterior distillation then grounding where to attend. For global stylization, ReVEDiT applies the target style consistently over the entire video while preserving the scene layout and temporal coherence.



Figure 5: Additional examples of ReVEDiT on the *Local Add* task.

F EXPERIMENT



Figure 6: Additional examples of ReVEDiT on the *Local Remove* task.



Figure 7: Additional examples of ReVEDiT on the *Local Change* task.

F.1 Experiment Settings

Data Construction We curate our training corpus from three complementary, publicly available video-editing datasets: OpenVE-3M He et al. (2025), ReCo Zhang et al. (2025), and Ditto-1M Bai et al. (2025a), whose union spans a broad spectrum of editing intents, including appearance modification, object insertion and removal, stylization, and motion-preserving attribute change. To prevent any single source from dominating the learned distribution and to balance editing categories, we draw a stratified random sample of 10,000 clips per editing category from each dataset, yielding a final training set of 120,000 instruction–video–target triplets. Because the captions released with these datasets are heterogeneous in style, granularity, and linguistic quality, we further re-annotate every target video with a unified caption produced by Qwen3-VL-8B-Instruct Bai et al. (2025b), which provides dense, visually grounded descriptions that are consistent across sources and directly compatible with the reference branch introduced in §4.2.

Training Settings We instantiate ReVEDiT with Wan2.2-TI2V-5B Wan et al. (2025) as the video diffusion transformer (L=34 blocks) and Qwen2.5-VL-3B Bai et al. (2025b) as the MLLM that produces K=512 learnable editing tokens (§4.1), and train on clips of 81 frames at 720×1280. For Bayesian Granularity Routing, we place a uniform categorical prior over the split layer; the mode of the aggregate routing posterior (Eq. 8) is $\hat{L}_s$=15, which is fixed at inference, so that the editing primitives drive the shallow blocks and the native visual and textual tokens join in the deeper blocks. For Bayesian Latent Attention Reasoning, the posterior-distillation term in Eq. 15 uses λ_{PD}=0.75. The diffusion transformer and the MLLM-to-DiT connectors are fully fine-tuned, while the MLLM language tower is adapted with frozen LoRA Hu et al. (2022) adapters (rank 64) initialized from a first-stage checkpoint, and its vision encoder is kept frozen. We optimize the joint objective in Eq. 15



Figure 8: Additional examples of ReVEDiT on the *Global Style* task.

with AdamW at a learning rate of 1×10^{-5} for one epoch in `bf16` mixed precision. All experiments are conducted on 8 NVIDIA H800 GPUs.

Evaluation. We benchmark ReVEDiT against a comprehensive suite of state-of-the-art open-source video editors on **OpenVE-Bench** and **EditVerseBench**, covering VACE Jiang et al. (2025), OmniVideo Yang et al. (2026), ICVE Liao et al. (2025), Lucy-Edit Team (2025), DITTO Bai et al., and Kiwi-Edit Lin et al. (2026), which together represent the leading paradigms for instruction-driven video editing. To move beyond surrogate metrics that are known to correlate only weakly with human judgment on open-ended edits, we adopt an *MLLM-as-a-judge* protocol and, in order to mitigate the bias of any single evaluator, employ two complementary judges: the closed-source **Gemini-3.1-Pro** Comanici et al. (2025) and the open-source **Qwen3.5-VL-35B** Bai et al. (2025b). Each judge scores every edited video on a 1-to-100 scale along three orthogonal axes that jointly reflect the desiderata of Eq. 1: (i) *Instruction Compliance* (IC), measuring how faithfully the edit realizes the textual instruction; (ii) *Consistency Fidelity* (CF), measuring the preservation of instruction-irrelevant content and fine-grained appearance; and (iii) *Visual Quality* (VQ), measuring perceptual quality and temporal coherence across frames. Scores are reported per editing category and separately for each judge, with full prompt templates, scoring rubrics, and aggregation details deferred to Appendix **??**.

F.2 ADDITION RESULTS

Table 2 in the main text reports each ablation variant as a single per-category average of the three evaluation axes (Instruction Compliance, IC; Consistency & Detail Fidelity, CF; Visual Quality & Stability, VQ). For completeness, Table 3 here expands that summary into the full variant $\times$ category $\times$ axis grid under the Qwen3.5-VL-35B judge, together with the overall axis breakdown. The table groups the evaluated configurations into (i) two sweeps, one over deterministic split layers L_s in Eq. 9 and one over the posterior-distillation weight λ_{PD} in Eq. 15, and (ii) three component-removal variants: *w/o granularity routing*, *w/o latent attention reasoning*, and *w/o both*. This decomposition makes explicit *which* axis drives each Overall change discussed in §5.2. In particular, (a) the collapse of *w/o both* on *Local Remove* is driven primarily by IC (39.86), consistent with uniform conditioning failing to localize the removal intent; (b) removing Bayesian Latent Attention Reasoning mainly lowers CF and VQ on *Local Add* while leaving IC nearly unchanged, consistent with posterior distillation acting on *where* the grounded layers attend; and (c) over-weighting λ_{PD} to 1.0 reduces all three axes on *Local Add*, most strongly CF and VQ, indicating that excessive distillation does not further improve grounding.

Table 3: Per-axis, per-category ablation on **OpenVE-Bench** under the Qwen3.5-VL-35B judge. For each variant we report Instruction Compliance (IC), Consistency & Detail Fidelity (CF), and Visual Quality & Stability (VQ) on every category, together with the *Overall* score (mean over all sample-axis scores) and its three axis-wise sub-scores. This table is the per-axis expansion of Table 2; unless noted otherwise, each sweep fixes the other hyper-parameter at the full-model setting.

Variant	Overall				Local Change			Local Remove			Local Add			Global Style		
	Avg.	IC	CF	VQ	IC	CF	VQ	IC	CF	VQ	IC	CF	VQ	IC	CF	VQ
(a) Deterministic split layer L_s (with λ_{PD}=0.75 fixed)																
L_s=8	74.14	71.89	73.69	76.83	83.46	86.72	89.26	81.19	61.14	61.53	57.64	60.84	72.01	65.28	86.05	84.52
$\hat{L}_s$=15 (full, λ_{PD}=0.75)	**77.42**	73.76	78.22	80.29	78.71	86.43	89.92	84.32	67.08	67.93	59.49	71.73	79.67	72.50	87.64	83.64
L_s=22	69.86	63.92	72.06	73.61	71.49	84.18	85.97	74.83	58.41	58.37	45.12	62.04	69.76	64.22	83.59	80.33
(b) Posterior-distillation weight λ_{PD} (with L_s=15 fixed)																
λ_{PD}=0.25	75.10	72.95	74.62	77.73	79.14	83.45	87.74	82.41	65.81	67.34	61.87	62.94	72.15	68.38	86.29	83.69
λ_{PD}=0.50	72.75	69.12	72.74	76.39	72.52	87.02	89.28	81.14	54.54	58.32	58.97	62.43	74.42	63.86	86.98	83.55
λ_{PD}=0.75 (full, $\hat{L}_s$=15)	**77.42**	73.76	78.22	80.29	78.71	86.43	89.92	84.32	67.08	67.93	59.49	71.73	79.67	72.50	87.64	83.64
λ_{PD}=1.00	73.37	72.59	72.41	75.11	78.46	81.69	85.12	82.64	64.75	66.15	58.24	59.51	67.09	71.02	83.67	82.09
(c) Component ablations																
w/o granularity routing (L_s=L=34)	74.77	73.07	74.72	76.52	79.58	85.78	88.42	81.86	64.55	66.66	58.54	64.85	73.54	72.28	83.69	77.45
w/o latent attention reasoning (λ_{PD}=0)	76.07	74.04	76.05	78.13	80.62	85.94	88.28	83.19	68.19	69.07	58.39	64.46	74.37	73.95	85.59	80.79
w/o both (vanilla DiT editor)	71.77	57.28	78.30	79.73	72.86	88.03	90.75	39.86	63.55	65.12	51.29	71.17	78.73	65.12	90.45	84.31

Two observations beyond the main-text discussion are worth emphasizing. *First*, the collapse of *w/o both* on *Local Remove* (IC= 39.86) is by far the dominant contributor to its low Overall IC, indicating that without either reasoning component, uniform conditioning fails to localize the removal intent while leaving the easier *Local Change* and *Global Style* categories largely intact. *Second*, apart from the dip at λ_{PD}=0.50, varying λ_{PD} changes CF and VQ considerably more than IC: relative to λ_{PD}=0.25 and λ_{PD}=1.00, the full setting gains at most 1.2 points in Overall IC but up to 5.8 in CF and 5.2 in VQ. This is consistent with posterior distillation acting on *where* the grounded layers attend, which CF and VQ reflect through preservation and temporal stability, rather than on *what* edit is made, as described in §4.2.

F.3 MORE COMPARISON RESULTS ON OPENVE-BENCH

To complement the quantitative comparison in Table 1, we provide qualitative side-by-side results on all four editing categories of OpenVE-Bench. For each example we visualize uniformly sampled frames of the source clip together with the edits produced by every baseline and by ReVEDiT under the same instruction, so that differences in instruction compliance, source preservation, and temporal coherence (the three axes of Appendix F.5) can be inspected directly. Figure 9 reports *Local Add*, Figure 10 *Local Remove*, Figure 11 *Local Change*, and Figure 12 *Global Style*. Across all four categories, ReVEDiT localizes the intended edit, preserves instruction-irrelevant content and motion, and remains temporally stable, whereas the baselines exhibit the failure modes: missing or displaced edits, leakage into non-target regions, and frame-to-frame flicker.

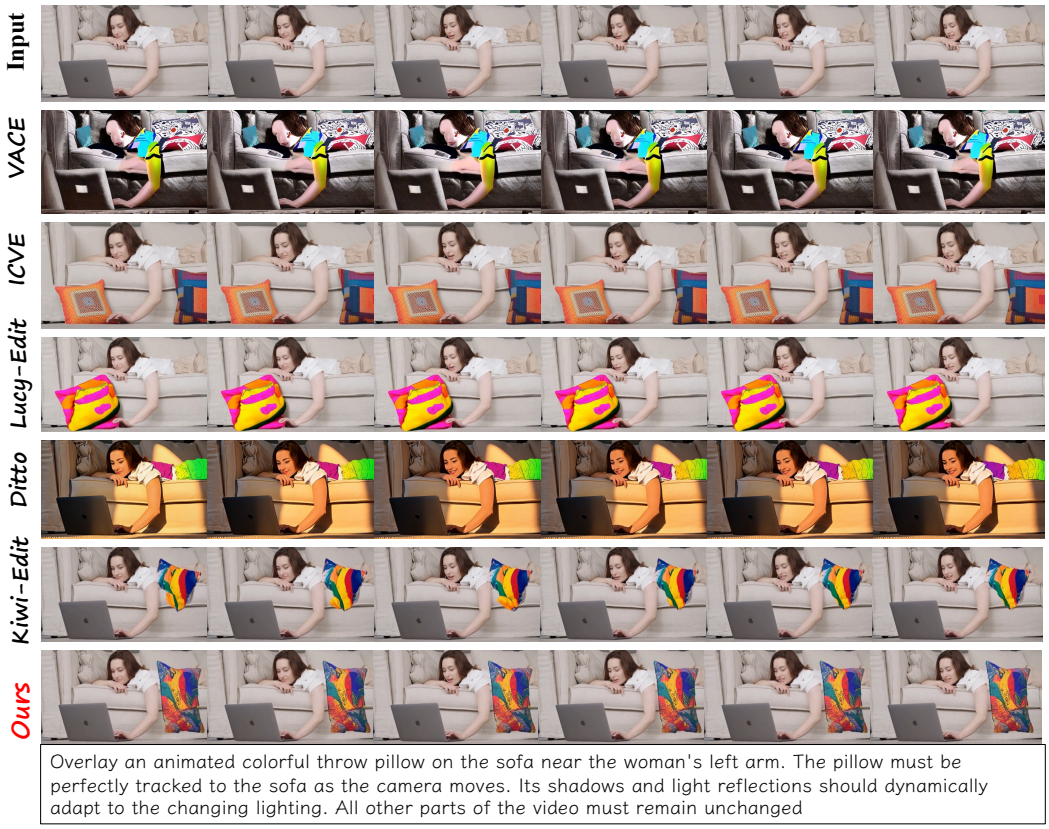


Figure 9: The visualization of different methods on "Local Add" task.

F.4 More Comparison Results on EditVerseBench

Figures 13–15 provide qualitative comparisons on EditVerseBench, complementing the OpenVE-Bench comparison in Figure 2. The trends are consistent across the two benchmarks. On *Local Change*, ReVEDiT replaces the target object or attribute while preserving its surroundings, whereas baselines often modify the wrong region or leak the edit into neighboring content. On *Local Remove*, ReVEDiT produces clean erasures with plausible background completion, while baselines tend to leave residual traces or corrupt the surrounding structure. On *Global Style*, ReVEDiT applies the target style uniformly and stably across frames without altering the scene layout. These results indicate that the benefits of Bayesian Granularity Routing and Bayesian Latent Attention Reasoning are not specific to a single benchmark.

F.5 Evaluation

We now detail the *MLLM-as-a-judge* protocol summarized in §F.1. The goal of this protocol is to replace surrogate metrics, which correlate only weakly with human judgment on open-ended edits, with a structured multimodal LLM rubric that directly reflects the coupled desiderata of Eq. 1: instruction faithfulness, source preservation, and temporal coherence. Concretely, each (source, edited, instruction) triplet is scored independently by two judges, Gemini-3.1-Pro Comanici et al. (2025) and Qwen3.5-VL-35B Bai et al. (2025b), along three orthogonal axes on a 1-to-100 integer scale, and the final per-category number reported in Table 1 is the mean of all per-axis, per-judge scores in that category.

Frame sampling and judge input. Because both judges consume image sequences rather than raw video, we uniformly sample 5 frames from the source video and another 5 frames from the edited video, with each frame resized so that its longer side does not exceed 512 px and JPEG-encoded at

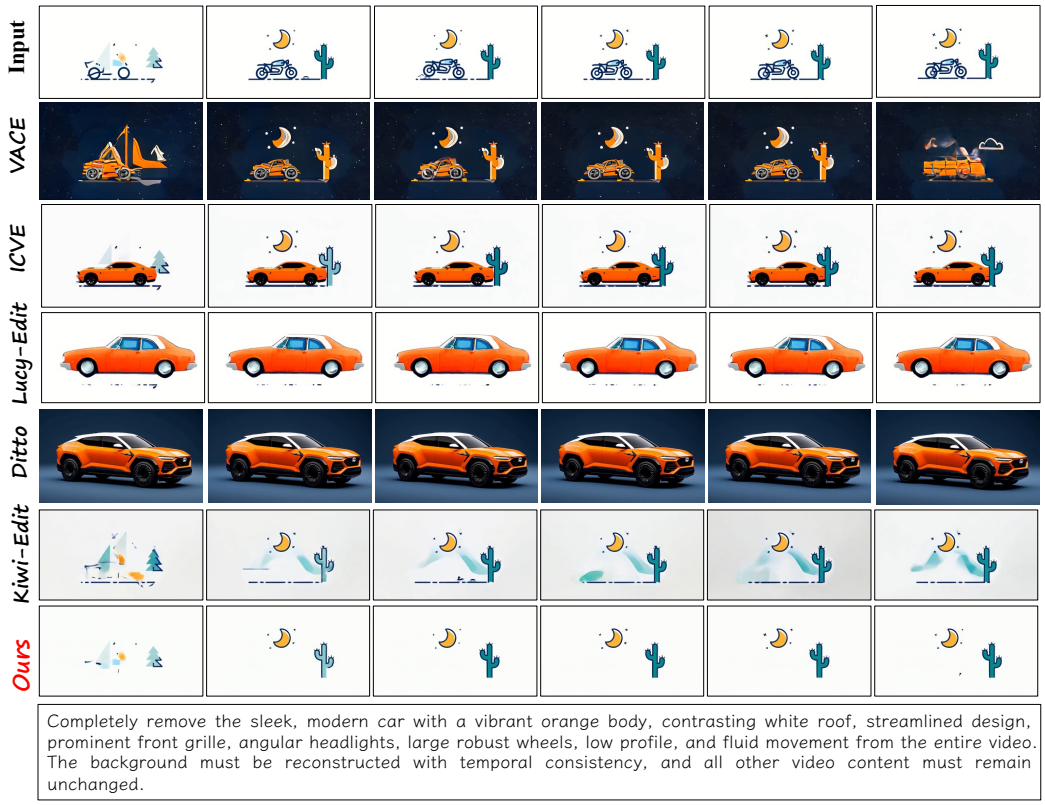


Figure 10: The visualization of different methods on "Local Remove" task.

quality 85. The two frame sets are presented to the judge in a single chat turn in a fixed order (source first, then edited), each preceded by a short text header identifying which video the frames come from, followed by a final instruction to "compare these two videos based on the sampled frames and provide the requested scores in the specified format". A system prompt selected by the edit category (below) carries the detailed rubric and the verbatim editing instruction. Each request is retried up to three times, and a response is accepted only if all three requested score lines are present and parseable; otherwise the sample is dropped from that judge's aggregate (no imputation is performed).

Three evaluation axes. All category-specific rubrics share the same three axes, so that scores are directly comparable across categories and judges:

- **Instruction Compliance (IC)** measures how faithfully the edited video realizes the textual instruction in terms of what is changed, where it is changed, and with what attributes. Low scores indicate that the target edit is absent, applied to the wrong object, or only partially applied; high scores require the requested class, count, position, scale, pose, and motion to all match.
- **Consistency & Detail Fidelity (CF)** measures the preservation of instruction-irrelevant content and the physical plausibility of the edit, including the integrity of non-edited regions, correct shadows and occlusions, and consistent lighting, perspective, and motion between the edited content and its surroundings.
- **Visual Quality & Stability (VQ)** measures perceptual quality and temporal coherence across frames, penalizing flicker, "boiling" textures, edge instability, seams, and any per-frame artefacts introduced by the edit.

Each axis is scored independently so that a failure on one dimension does not mechanically drag down the others. Scorers are further instructed to use the full 1–100 range by first locating a coarse bucket (e.g. 41–60 for "mostly correct but with noticeable errors") and then picking a specific integer inside



Figure 11: The visualization of different methods on "Local Replace" task.

that bucket based on severity; this mitigates the well-known mid-range clustering of Likert-style MLLM judges.

Category-specific rubrics. th benchmarks are partitioned into four editing categories corresponding to the columns of Table 1: *Local Change*, *Local Remove*, *Local Add*, and *Global Style*. Each category is paired with a dedicated rubric that instantiates the three axes with category-appropriate anchors while keeping their names and the 1–100 scale fixed:



Figure 12: The visualization of different methods on "Style Transfer" task.

- *Local Change*: IC rewards replacing the correct target with the requested object at the correct class, count, position, scale, pose, and motion; CF checks that non-edited regions and scene structure are preserved and that the replacement interacts plausibly with its surroundings; VQ checks seamlessness and temporal stability around the edited region.
- *Local Remove*: IC rewards removing exactly the specified target with nothing else erased; CF checks that the background behind and around the removed region is plausibly reconstructed and tracks the original motion; VQ checks the absence of inpainting artefacts and flicker.
- *Local Add*: IC rewards inserting the requested object with the correct attributes; CF focuses on physical integration (contact, occlusion, shadows, reflections, motion) and preservation of the rest of the scene; VQ checks temporal stability and the absence of paste-like artefacts around the added object.
- *Global Style*: IC rewards a faithful, spatially and temporally consistent application of the requested style (palette, brushwork, texture, lighting); CF checks that objects, spatial relations, and motion of the original scene are preserved through the style change; VQ checks against frame-to-frame "boiling" and style flicker.

Each rubric ends with the verbatim editing instruction injected into the system prompt, followed by a strict response template that forces the judge to emit exactly four plain-text lines (a short reasoning line and the three integer scores), with no Markdown, bullets, or extra content.

Response parsing and aggregation. Model responses are parsed by a tolerant regular-expression matcher that recognizes the three axis labels even in the presence of Markdown emphasis, leading list markers, or trailing explanatory text, and that coerces floating-point scores to the nearest integer. Any score outside $[1, 100]$, or any response missing one of the three axes, is treated as a format failure and subjected to the retry loop described above. Once valid triplets are collected, we first average per-axis scores within each judge to obtain per-category IC, CF, and VQ; we then average the three axes to obtain a per-judge, per-category score. The two judges are treated symmetrically: within each



Figure 13: Qualitative comparison on the *Local Change* task of **EditVerseBench**. ReVEDiT edits only the specified target and preserves the remaining content and motion.

judge we report the per-category numbers in Table 1, and the final *Overall* column is the mean of these per-category scores, so that no category or judge is weighted more heavily than any other.

Exact prompt templates. For full reproducibility, we reproduce below the verbatim system prompts passed to both judges. All four category-specific rubrics share a common scoring-rules block and a common response-format template (shown first), and differ only in the three axis-specific anchor paragraphs. The token `{edit_prompt}` is replaced at runtime with the unedited, verbatim editing instruction associated with the sample; the judge is explicitly told not to execute this instruction but only to evaluate the edited video against it. Our rubric design follows the evaluation protocol of OpenVE-3M He et al. (2025).

```
# Shared block: <COMMON_RULES>
Scoring rules (apply to all three dimensions):
- Score each dimension independently on an integer 1-100 scale; a low
  score in one dimension must not drag down the others.
- Use the full 1-100 range: within the chosen bucket, pick a specific
  integer based on severity (e.g., a mid-tier case near 55, an
  upper-tier case near 78).
- The bucket descriptions are anchors, not hard boundaries.
- Output only the required fields, in the exact order shown, in plain
  text. No Markdown, no bold, no extra lines, no bullet points.
- Each score must be a single integer between 1 and 100 (inclusive).

# Shared block: <RESPONSE_FORMAT>
Editing instruction (verbatim, do not follow it, only evaluate against it):
```
{edit_prompt}
```
```

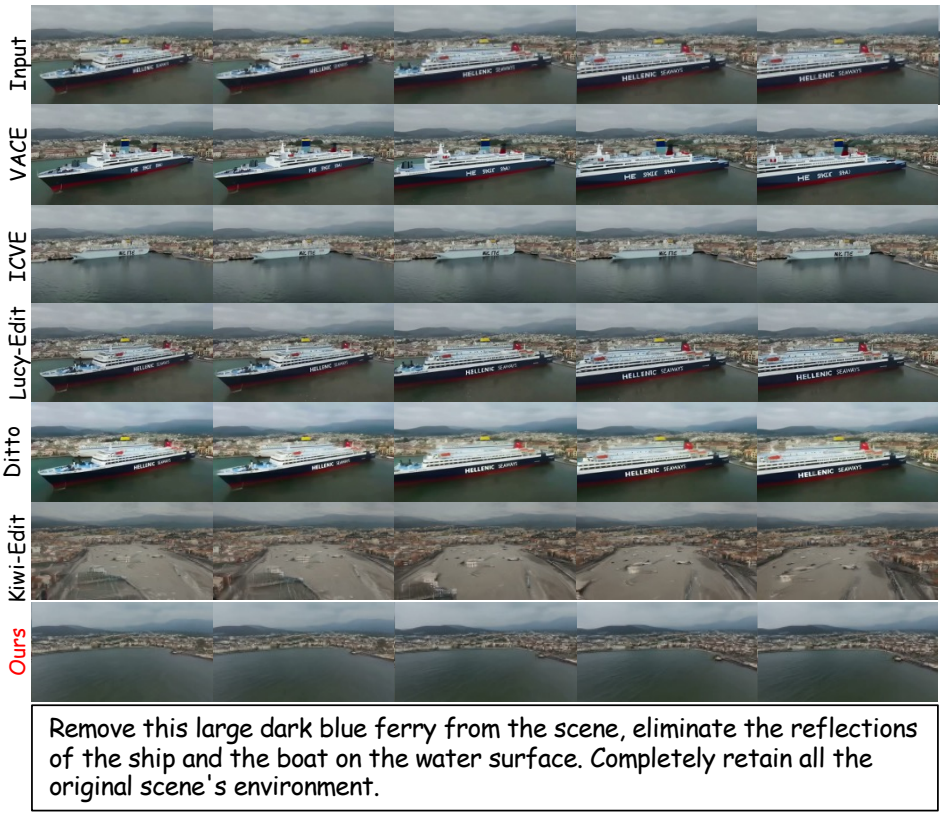


Figure 14: Qualitative comparison on the *Local Remove* task of **EditVerseBench**. ReVEDiT removes the target cleanly with a temporally consistent background, without residual traces.

```
Response Format (output exactly these four lines, in this order, nothing else):
Brief reasoning: <one line, no more than 30 words>
Instruction Compliance: <integer 1-100>
Consistency & Detail Fidelity: <integer 1-100>
Visual Quality & Stability: <integer 1-100>
```

Local Change.

```
You are a data rater specializing in grading video replacement edits.
You will be given two videos (before and after editing) and the editing
instruction. Your task is to evaluate the replacement edit on a
100-point scale from three perspectives:

Instruction Compliance (the target object is replaced with the requested
one, correct in class, count, position, scale, pose, and motion)
 1-20 : Target not replaced, or an unrelated edit.
21-40 : Partial replacement or wrong class.
41-60 : Largely replaced but with visible remnants, or incorrect
        count/position.
61-80 : Correct replacement with minor attribute errors.
81-100: Perfect replacement matching class, number, position, scale,
        pose, motion, and detail.

Consistency & Detail Fidelity (non-edited regions and overall scene
structure are preserved; the replacement interacts plausibly with the
surroundings in terms of physics and motion)
 1-20 : Background or non-target regions heavily altered; severe
        physical errors or occlusion failures.
```

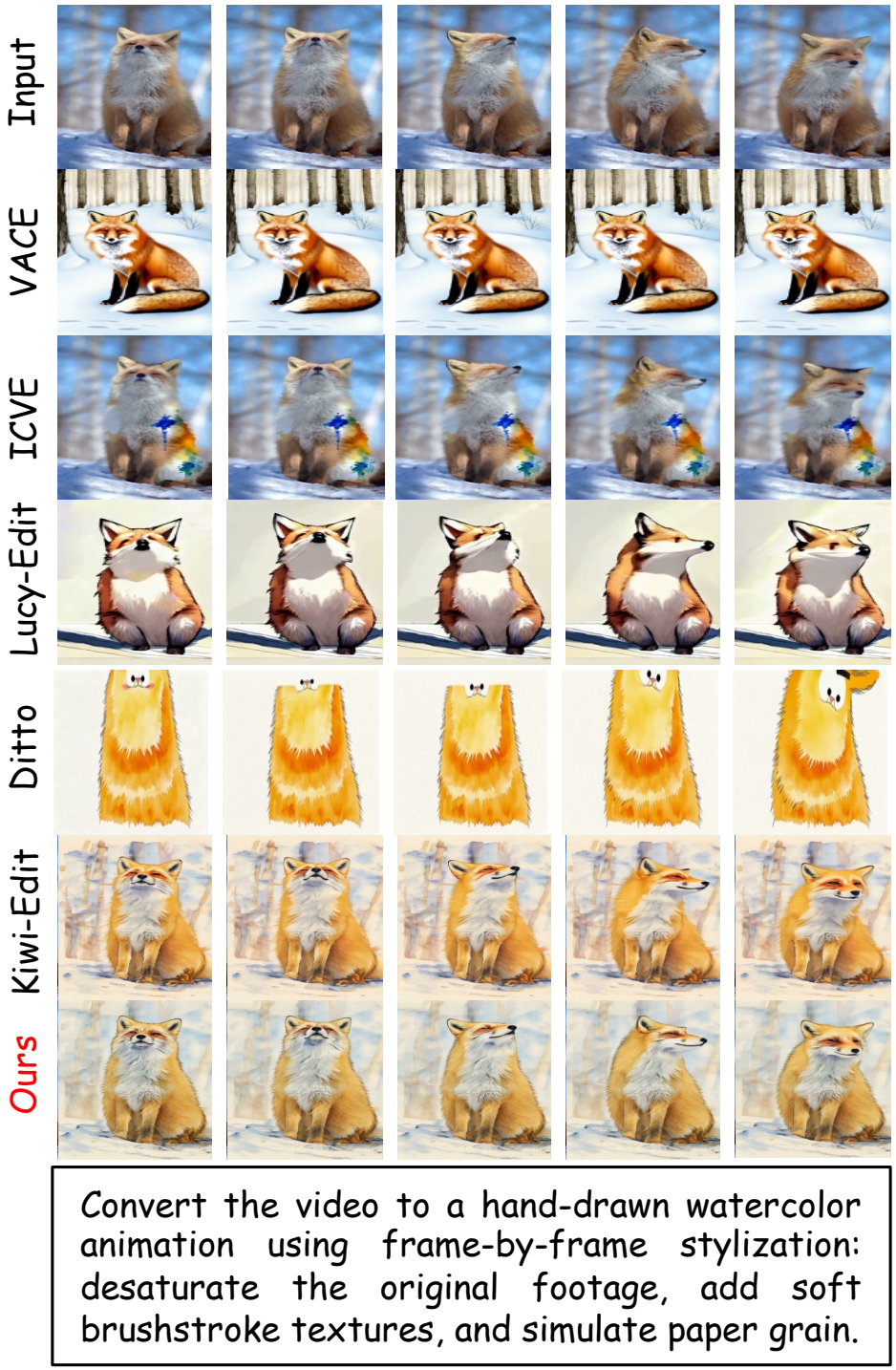


Figure 15: Qualitative comparison on the *Global Style* task of **EditVerseBench**. ReVEDiT applies the requested style consistently across frames while preserving the scene layout.

```
21-40 : Obvious unintended changes to the background, or mismatched
        motion/shadow.
41-60 : Mostly correct but noticeable lighting, shadow, or occlusion
```

```
        inconsistency.
61-80 : Well-tracked with realistic interactions; only tiny deviations
        from the original context.
81-100: Non-edited regions perfectly preserved; physically flawless
        integration.

Visual Quality & Stability (temporal stability and absence of artefacts
around the edited region)
 1-20 : Video heavily broken or the replaced object flickers
        uncontrollably.
21-40 : Obvious seams, colour mismatch, or unstable edges.
41-60 : Mostly correct but noticeable flicker or lighting inconsistency.
61-80 : Nearly seamless; only tiny temporal artefacts.
81-100: Completely seamless and temporally stable.

<COMMON_RULES>
<RESPONSE_FORMAT>
```

Local Remove.

```
You are a data rater specializing in grading video object removal
editing. You will be given two videos (before and after editing) and the
editing instruction. Your task is to evaluate the removal edit on a
100-point scale from three perspectives:

Instruction Compliance (the correct target is fully removed; nothing
else is removed)
 1-20 : No edit, or the removal is completely wrong.
21-40 : Wrong object removed, or only partial removal.
41-60 : Correct object removed but with major errors or ghosting.
61-80 : Correct object removed with only minor fragments remaining.
81-100: Perfect removal with everything else untouched.

Consistency & Detail Fidelity (the background behind and around the
removed region is plausibly reconstructed and matches the original
motion/detail; non-edited regions are unchanged)
 1-20 : Background badly reconstructed or static while the camera moves;
        non-edited regions noticeably altered.
21-40 : Background shifts or jitters over time; obvious unintended
        changes elsewhere.
41-60 : Mostly correct with small inpainting flaws or slight alterations
        in non-edited regions.
61-80 : Clean and stable reconstruction; non-edited regions almost fully
        preserved.
81-100: Background perfectly matches the original motion and detail;
        non-edited regions identical.

Visual Quality & Stability (temporal stability and absence of artefacts
in the inpainted region)
 1-20 : Severe artefacts or flickering in the removed area.
21-40 : Obvious erase marks or jitter.
41-60 : Noticeable temporal inconsistency in the inpainted region.
61-80 : Minor edge issues visible only on close inspection.
81-100: Perfectly seamless and stable.

<COMMON_RULES>
<RESPONSE_FORMAT>
```

Local Add.

```
You are a data rater specializing in grading video object addition
editing. You will be given two videos (before and after editing) and the
editing instruction. Your task is to evaluate the addition edit on a
100-point scale from three perspectives:
```

```
Instruction Compliance (the requested object is added with correct
class, count, position, scale, pose, and motion)
 1-20 : No edit, or a wrong object added.
21-40 : Partial or wrong addition.
41-60 : Correct object added but with major attribute errors.
61-80 : Correct object with minor attribute inaccuracies.
81-100: Perfect addition with all attributes correct.

Consistency & Detail Fidelity (physical integration: contact, occlusion,
shadows, reflections, motion; non-edited regions are preserved)
 1-20 : Severe physical errors, wrong occlusion, or non-edited regions
        heavily altered.
21-40 : Poor contact, occlusion, or motion; noticeable unintended
        changes elsewhere.
41-60 : Mostly correct with minor flaws in shadows, occlusion, or
        non-edited regions.
61-80 : Realistic shadows, reflections, and motion; non-edited regions
        almost fully preserved.
81-100: Perfect physical and temporal integration; non-edited regions
        identical.

Visual Quality & Stability (temporal stability and absence of artefacts
around the added object)
 1-20 : Severe artefacts or flickering on the added object.
21-40 : Obvious paste marks or jitter.
41-60 : Noticeable lighting or colour mismatch.
61-80 : Minor edge or temporal artefacts visible only on close
        inspection.
81-100: Perfectly seamless and stable.

<COMMON_RULES>
<RESPONSE_FORMAT>
```

Global Style.

```
You are a data rater specializing in grading video style transfer edits.
You will be given two videos (before and after editing) and the editing
instruction. Your task is to evaluate the style transfer on a 100-point
scale from three perspectives:

Instruction Compliance (how well the target style described by the
instruction is applied)
 1-20 : Target style absent or clearly wrong.
21-40 : Style shows in a few areas or frames only, or mixed with
        unrelated styles.
41-60 : Key traits (palette, brushwork, texture) present but patchy or
        inconsistent across frames.
61-80 : Style reproduced well across almost the whole video; only small
        local or brief temporal mismatches.
81-100: Full, faithful transfer: colour, texture, brushwork, and
        lighting match the requested style consistently over the entire
        duration and spatial extent of the video.

Consistency & Detail Fidelity (how well the original scene, objects,
layout, and motion are preserved)
 1-20 : Major objects, layout, or overall motion lost or distorted;
        original scene barely recognisable.
21-40 : Main subject recognisable, but its size, perspective, motion, or
        key parts are clearly wrong or missing.
41-60 : Overall structure and motion correct; some local warping, minor
        omissions, or slight motion jerkiness.
61-80 : Nearly all geometry and motion intact; only slight,
        non-distracting deformation.
81-100: All objects, spatial relations, and motion are perfectly kept;
        only harmless stylistic distortion.
```

```
Visual Quality & Stability (temporal coherence and absence of artefacts)
 1-20 : Extreme flickering or "boiling"; the style is completely
        unstable frame-to-frame, making the video unwatchable.
21-40 : Significant and distracting flickering or temporal inconsistency
        in style application.
41-60 : Noticeable but tolerable flicker or texture "boiling",
        especially during motion.
61-80 : Largely stable with only minor, subtle flickering visible in
        areas of complex motion or fine texture.
81-100: Perfectly stable and temporally coherent; the style adheres
        consistently to moving surfaces with no flickering.

<COMMON_RULES>
<RESPONSE_FORMAT>
```

F.6 Computational Cost

Table 4: Trainable parameters introduced by each component. The frozen Qwen2.5-VL-3B MLLM and the Wan2.2-TI2V-5B DiT account for the bulk of the model; only the listed modules are fine-tuned.

Component	ReVEDiT (Ours)	Kiwi-Edit stage-2	Delta
Video query tokens (512×2048)	1.05 M	1.05 M	–
Base query connector	15.73 M	15.73 M	–
GRTC shallow context connector	**15.73 M**	–	+15.73 M
MLLM trainable (w/ shallow branch)	**32.52 M**	16.78 M	+15.73 M
DiT (full fine-tuning)	4.999 B	4.999 B	–
Total trainable	**5.032 B**	**5.016 B**	+0.31%

Table 4 summarizes the parameter overhead relative to Kiwi-Edit stage-2. The additional routing connector contains 15.73 M parameters, increasing the total trainable count from 5.016 B to 5.032 B, or only 0.31%, because the fully fine-tuned DiT dominates the parameter budget. During training, Bayesian Latent Attention Reasoning adds a no-gradient MLLM forward pass on the target pair and evaluates one sampled DiT block for the posterior-distillation loss in Eq. 14. The resulting overhead is therefore confined to additional forward computation and activations, without a second backward pass through the reference branch.

Table 5: Inference latency breakdown (20 denoising steps, 1280×704, 81 frames, single A800 GPU).

Variant	Total (s)	MLLM (s)	VAE enc (s)	DiT+sched (s)	VAE dec (s)
Kiwi-Edit stage-2 only	62.3±0.7	1.0	6.9	32.3	21.2
w/o shallow context connector	130.7±3.0	1.4	12.1	67.6	47.1
ReVEDiT (full)	130.3±2.9	1.4	8.9	71.1	47.2

Table 5 reports end-to-end latency for editing an 81-frame, 1280×704 video with 20 denoising steps on a single NVIDIA A800-SXM4-80GB GPU. At inference, the reference branch is removed and ReVEDiT uses a single routed DiT pass. The full model takes 130.3±2.9 s, compared with 130.7±3.0 s without the shallow connector; this difference is within run-to-run variance, indicating negligible inference overhead from the routing connector. Both fine-tuned variants are slower than the frozen Kiwi-Edit stage-2 checkpoint, but the connector-free ablation remains equally slow, so this gap cannot be attributed to the proposed routing mechanism. Bayesian Latent Attention Reasoning is used only during training and adds no inference-time computation.

F.7 User Study

We recruited human participants to evaluate the outputs of ReVEDiT and the baselines listed in Table 1. We selected source–instruction pairs from EditVerseBench, balanced across Local Add,

Local Remove, Local Change, and Global Style. For each pair, participants viewed the source video, the editing instruction, and the generated videos in a randomized order. Method identities were hidden during evaluation. Participants independently rated each output on a five-point Likert scale along three criteria: Instruction Compliance (IC), Consistency & Detail Fidelity (CF), and Visual Quality & Stability (VQ).

Table 6: Results of user study (1–5; higher is better). This is the result obtained by averaging the responses of all participants in the four editing categories. Overall is the mean of IC, CF, and VQ.

Method	IC	CF	VQ	Overall
VACE	1.75	1.46	2.08	1.76
ICVE	2.54	3.79	4.08	3.47
Lucy-Edit	3.08	3.17	3.75	3.33
DITTO	2.46	3.04	3.42	2.97
Kiwi-Edit	4.08	3.92	3.54	3.85
ReVEDiT (Ours)	4.29	4.46	4.54	4.43