Transforming Image Editors into Video Editors

Feng Wang[1,2*], **Zijie Li**[1], **Ceyuan Yang**[1], **Alan Yuille**[2], **Peng Wang**[1]
[1]ByteDance Seed [2]Johns Hopkins University

Abstract

Recent image editing systems have achieved impressive semantic understanding, visual fidelity, and instruction-following ability, while video editing remains substantially more difficult and costly. In this paper, we present a simple alternative to end-to-end video editing: instead of training a monolithic video editor, we transform a strong image editor into a video editor through anchor-based generation. Our key insight is that video editing can be decomposed into two subproblems: editing a sparse set of keyframes and propagating those edits across time. Based on this observation, we propose Anchor-based Video Editing (AVE), a two-stage framework in which a powerful image editor first performs composed editing on selected keyframes, and a motion-guided image-to-video diffusion model then generates the final video by treating the edited keyframes as fixed anchors. This design directly inherits the strengths of modern image editors while avoiding expensive end-to-end video editing training. Experiments on IVEBench and VIE-Bench show that AVE achieves strong performance in instruction following, temporal consistency, and content fidelity. Further ablations reveal that final video editing quality is strongly correlated with the quality of the image editor, suggesting that future progress in video editing may come from stronger image editing foundations and lightweight transfer to video. Code is available at https://github.com/wangf3014/AVE.

1 Introduction

Recent advances in image editing have pushed the field to a remarkable level of maturity. Modern image editing systems now exhibit strong semantic understanding, high visual fidelity, and precise instruction-following ability, often reaching a quality level suitable for real-world applications (OpenAI, 2025; Google, 2025; Deng et al., 2025; Qwen Team, 2025; Brooks et al., 2023; Black Forest Labs, 2025; Team Seedream et al., 2025; Liu et al., 2025; Xiao et al., 2025). In contrast, video editing still lags noticeably behind. Compared with image editing, video editing models must handle much longer spatiotemporal contexts, which makes them significantly more expensive to train. Meanwhile, collecting high-quality video editing data is also considerably harder, due to the need for temporal consistency, motion diversity, and reliable instruction alignment across frames. As a result, while image editing systems have rapidly advanced to a high level of quality and controllability, mainstream video editing models still remain clearly behind in visual quality, semantic consistency, and instruction-following performance (Kling Team et al., 2025; Jiang et al., 2025; Bai et al., 2026; Cheng et al., 2024; Wu et al., 2025; Lin et al., 2026; Rao et al., 2026; Zhang et al., 2026a; Runway, 2025; Pika, 2026; Zi et al., 2025; Li et al., 2025; Tan et al., 2025; DecartAI Team, 2025; Wei et al., 2026; Mou et al., 2025; Liao et al., 2025; He et al., 2026; Chen et al., 2026a; Zhang et al., 2026b).

To narrow this gap, this paper studies how to transfer the editing capability of image systems to video models at minimal cost. Through a series of studies, we surprisingly find that this transfer is much simpler than it appears to be. In particular, although video editing is a 3D video-to-video task, its primary challenge remains the underlying 2D image-to-image editing problem. By contrast, handling temporal information, such as preserving or modifying motion, can be achieved with relatively little

*Work done when interning at ByteDance Seed.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Paradigm	Consistency	Frame Quality	Instruction Following	Training Cost
End-to-end video editing	Good ✅	Good ✅	Depends on training 🤔	Expensive 💸
Per-frame editing	Bad ❌	Excellent 🌟	Bad ❌	Cheap 💰
Two-stage editing of AVE (ours)	Good ✅	Excellent 🌟	Great and robust 🌟	Normal 👌

Table 1: ***Intuitive comparison of three video editing paradigms.*** Our model attains favorable video consistency, frame-wise quality and instruction-following abilities with a normal cost.

additional training cost. Motivated by this observation, we introduce **A**nchor-based **V**ideo **E**diting (**AVE**), a new instruction-guided video editing paradigm which explicitly decomposes the task into two stages: *Composed Image Editing* and *Keyframe Interpolation*.

Specifically, given an input video, we first extract a sparse set of keyframes and jointly edit them with a powerful image editing model. By performing composed image editing over multiple keyframes, we preserve scene-level and semantic consistency across the edited frames while fully benefiting from the strong instruction-following and visual generation capabilities of modern image editors. We then use an image-to-video diffusion model (*e.g.,* Wan2.2-I2V (Wan Team et al., 2025)) to generate the target video, where the edited keyframes serve as fixed anchors and the generation is constrained by the motion representation of the input video. In this way, the model is required to preserve the edited content at key moments while propagating the edits smoothly across intermediate frames. This decomposition provides a simple and scalable path for transforming image editors into video editors, allowing video editing systems to directly inherit the rapid progress of image editing models without requiring expensive end-to-end video editing training.

Compared with the common end-to-end video editing frameworks (*e.g.*, VACE (Jiang et al., 2025)) and the most naive per-frame editing strategy, the main advantages of AVE can be summarized as follows. First, the image editing model in Stag-1 is fully decoupled from Stage-2, which allows AVE to directly leverage any mature image editor to achieve the best possible frame-wise editing quality. As a result, AVE can seamlessly inherit the rapid progress of image editing systems, including stronger semantic understanding, better visual fidelity, and more accurate instruction following, without any modification to the image editor itself. Second, unlike naive per-frame editing, which often suffers from severe flickering and temporal inconsistency, AVE performs interpolation between edited keyframes using a video diffusion model, enabling smooth propagation of edits across intermediate frames while maintaining strong temporal consistency. Third, AVE is highly cost-efficient to train. In contrast to end-to-end video editing frameworks that require substantial training resources, AVE only needs to train a lightweight motion encoder, while keeping all other components fixed and initialized from pretrained models. In our study, we find that a small motion encoder and a relatively short context length are already sufficient to capture the motion information needed for high-quality video editing. The comparison is summarized in Table 1.

Extensive experiments validate the effectiveness of AVE from multiple perspectives. On IVEBench (Chen et al., 2026b), AVE achieves the best performance on both the short and long subsets across all reported metrics, showing that the proposed decomposition leads to a strong and well-balanced video editing capability rather than improving only a single aspect of the task. In particular, AVE is especially strong in motion-related metrics, which supports our design choice of using a motion-guided video interpolation stage to propagate edits smoothly while preserving the original temporal dynamics. AVE also obtains clear gains in semantic consistency and content fidelity, suggesting that high-quality edited anchors provide a reliable foundation for long-range video generation. On VIE-Bench (Mou et al., 2025), AVE generalizes well across diverse editing types, including add, swap, remove, and style transfer, and remains competitive even against strong closed-source systems. In addition, our ablations reveal two important findings: first, the final video editing quality is strongly correlated with the quality of the Stage-1 image editor, confirming that AVE effectively transfers image editing ability into the video domain; second, AVE is robust to different keyframe selection strategies, with a moderate keyframe budget already sufficient to achieve strong performance. These results consistently support our central claim that, once high-quality edited anchors are available, the remaining image-to-video transfer is much simpler than end-to-end video editing would suggest. We hope this study provides useful insights for future video editing systems.

2 Related Work

Instruction-based image editing. Instruction-based image editing has advanced rapidly, evolving from early diffusion-based editors such as InstructPix2Pix (Brooks et al., 2023) and InstructEdit (Wang et al., 2023a) to more capable multimodal systems such as SmartEdit (Huang et al., 2024),

OmniGen (Xiao et al., 2025), Step1X-Edit (Liu et al., 2025), and BAGEL (Deng et al., 2025). Recent open and commercial image systems, including Qwen-Image-Edit (Qwen Team, 2025), FLUX.1 Kontext (Black Forest Labs, 2025), and Nano Banana 2 (Fortin and Alessio, 2026), further improve semantic understanding, visual fidelity, and instruction following. These advances suggest that high-quality editing is increasingly solved in the image domain. Our work is complementary to this line of research: instead of proposing another image editor, we study how to transfer the editing capability of mature image systems into video models with minimal additional cost.

Instruction-guided video editing. Text-guided video editing is substantially more challenging because it must preserve temporal consistency while executing semantic edits. Early methods mainly adapt pretrained image diffusion models through inversion, one-shot tuning, or feature propagation, such as Tune-A-Video (Wu et al., 2023), FateZero (Qi et al., 2023), TokenFlow (Geyer et al., 2024), and InstructVideo (Yuan et al., 2024). More recent feed-forward approaches rely on synthetic datasets and end-to-end training, including InsV2V (Cheng et al., 2024), InsViE (Wu et al., 2025), Ditto (Bai et al., 2026), VACE (Jiang et al., 2025), Kiwi-Edit (Lin et al., 2026), InsEdit (Rao et al., 2026), and SAMA (Zhang et al., 2026a). Concurrent efforts also explore unified or in-context formulations and larger-scale supervision, such as ICVE (Liao et al., 2025), ReCo (Zhang et al., 2026b), and OpenVE-3M (He et al., 2026). While these methods have significantly improved the field, they typically require substantial video-specific training data and computation, and their editing quality still trails the best image editors. In contrast, our method does not treat video editing as a monolithic end-to-end problem. Instead, it explicitly decomposes the task into composed keyframe editing and temporal interpolation, so that the hardest part of the problem can be delegated to a strong image editor.

Image-to-video generation and motion modeling. A related line of work studies image-to-video generation and controllable motion synthesis. Representative approaches include AnimateDiff (Guo et al., 2024), VideoCrafter2 (Chen et al., 2024), VideoComposer (Wang et al., 2023b), and methods that explicitly decouple content and motion (Shen et al., 2024). These models show that motion can often be modeled separately from appearance, and that pretrained image generators can be extended to videos with relatively lightweight temporal modules. AVE builds on this insight but applies it to editing rather than pure generation. Instead of asking a video model to jointly solve semantic editing and temporal reasoning, we use the video model mainly to interpolate between edited keyframe anchors under the motion cues of the source video. This design allows AVE to inherit the rapid progress of image editing models while keeping video-specific training lightweight.

3 Method

3.1 Overview

As illustrated in Figure 1, given an input video $V = \{x_t\}_{t=1}^{T}$ and a text editing instruction y, our goal is to generate an edited video $\hat{V} = \{\hat{x}_t\}_{t=1}^{T}$ that faithfully follows the instruction while preserving the source motion and temporal coherence. Our key observation is that video editing can be decomposed into two largely independent subproblems: *(i)* high-quality semantic editing on a sparse set of keyframes, and *(ii)* temporally coherent interpolation between the edited keyframes. Let $\mathcal{A} = \{a_1, \ldots, a_K\}$ denote the indices of K selected keyframes from the source video. AVE first edits the keyframe set $\mathbf{x}_{\mathcal{A}} = \{x_a\}_{a \in \mathcal{A}}$ with an image editing model $\hat{\mathbf{x}}_{\mathcal{A}} = \mathcal{E}(\mathbf{x}_{\mathcal{A}}, y)$, and then uses a video diffusion model to generate the final video conditioned on the edited keyframes as fixed anchors and a lightweight motion representation extracted from the source video $\hat{V} = \mathcal{G}(\hat{\mathbf{x}}_{\mathcal{A}}, \mathbf{m}(V))$. A key design of AVE is that Stage-1 and Stage-2 are fully decoupled: the only interface between them is the edited keyframes. This allows us to improve the frame-wise editing quality by simply replacing the image editor, without retraining the video generator.

3.2 Stage-1: Composed Image Editing

Training a multi-image editor from video data. For open-source instantiations, we train a composed image editor by adapting BAGEL (Deng et al., 2025) with fine-tuning. Starting from a private video editing dataset of triplets $(V, y, V^\star)$, where $V^\star$ is the edited target video, we construct multi-image editing training tuples by sampling aligned keyframes from the source and target videos $(\mathbf{x}_{\mathcal{A}}, y, \mathbf{x}^\star_{\mathcal{A}})$,

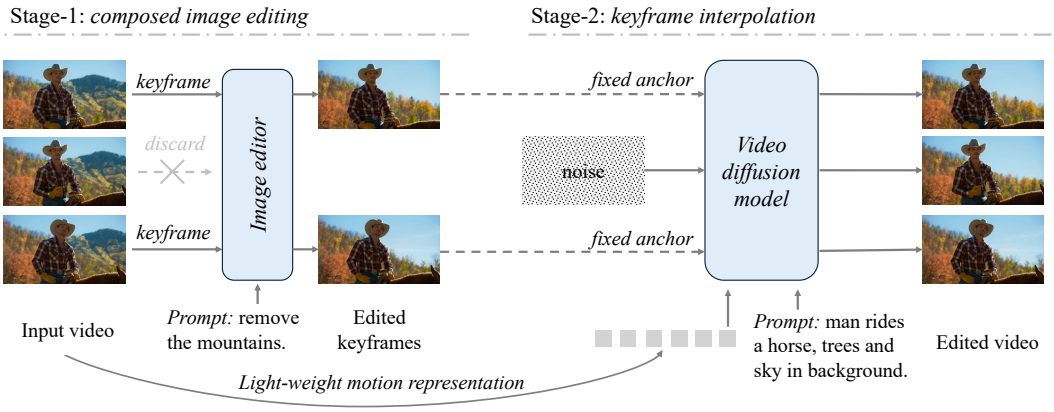

Figure 1: ***Overview of our two-stage pipeline.*** In Stage-1, we edit a sparse set of keyframes using either open-source or API-based image editors. In Stage-2, an image-to-video diffusion model interpolates between the edited anchors while preserving the source motion via a lightweight motion encoder and anchor clamping.

where $\mathbf{x}^\star_{\mathcal{A}} = \{x^\star_a\}_{a\in\mathcal{A}}$ denotes the target keyframes corresponding to the same timestamps. In total, we build approximately 1.5M multi-image editing examples from private video editing data.

Let $\mathcal{V}_{\text{img}}$ be the image VAE used by the editor, and let $\mathbf{z}_{\mathcal{A}} = \mathcal{V}_{\text{img}}(\mathbf{x}_{\mathcal{A}})$, $\mathbf{z}^\star_{\mathcal{A}} = \mathcal{V}_{\text{img}}(\mathbf{x}^\star_{\mathcal{A}})$ be the source and target latent codes. We fine-tune BAGEL with the standard diffusion denoising objective:

$$\mathcal{L}_{\text{img}} = \mathbb{E}_{\mathbf{z}^\star_{\mathcal{A}},\,\epsilon,\,\tau}\left[\left\|\epsilon - \epsilon_\phi(\alpha_\tau \mathbf{z}^\star_{\mathcal{A}} + \sigma_\tau \epsilon,\ \mathbf{z}_{\mathcal{A}},\ y,\ \tau)\right\|_2^2\right], \tag{1}$$

where ϵ_ϕ is the denoiser of the fine-tuned BAGEL, and $(\alpha_\tau, \sigma_\tau)$ are the diffusion coefficients at timestep τ. By training on *multiple* aligned frames jointly, the editor learns to perform consistent edits across keyframes, which is critical for downstream video generation.

Closed-source alternative. While the above open-source instantiation is useful for controlled experiments, our framework is not tied to any particular image editor. In practice, unless otherwise specified, we instantiate Stage-1 with the latest Nano Banana API (Fortin and Alessio, 2026) to maximize editing quality. For strong closed-source models, we find that the most effective strategy is to edit all selected keyframes within a single multi-turn conversation, which naturally preserves subject identity, style, and scene-level consistency across frames. In the ablation study, we compare different API-based editors as well as open-source image editors within the same Stage-2 pipeline.

3.3 Stage-2: Keyframe Interpolation

Stage-2 takes the edited keyframes from Stage-1 and generates a temporally coherent target video using a pretrained image-to-video model. We use Wan2.2-I2V-A14B (Wan Team et al., 2025) as the backbone and keep all its pretrained components frozen, except for a lightweight motion encoder and its associated control branch.

Lightweight motion encoder. Let $\mathcal{V}^{\text{enc}}_{\text{3D}}$ and $\mathcal{V}^{\text{dec}}_{\text{3D}}$ denote the 3D VAE encoder and decoder of Wan2.2-I2V-A14B. The source video is encoded into latent space as $Z = \mathcal{V}^{\text{enc}}_{\text{3D}}(V) \in \mathbb{R}^{T\times H\times W\times C}$. We project latent tokens into the motion embedding space: $U^{(0)}_{t,h,w} = W_{\text{in}} Z_{t,h,w} + e^{\text{temp}}_t + e^{\text{spatial}}_{h,w}$, where $W_{\text{in}} \in \mathbb{R}^{C\times d}$ is a learnable linear projection, and e^{temp}_t and $e^{\text{spatial}}_{h,w}$ denote temporal and spatial positional embeddings, respectively. The resulting tensor $U^{(0)} \in \mathbb{R}^{T\times H\times W\times d}$ is then processed by a 6-layer factorized spatiotemporal Transformer. Each layer consists of a spatial self-attention block followed by a temporal self-attention block and an MLP. We inject the final motion representation into the frozen Wan2.2 DiT through a ControlNet-style residual branch (Zhang et al., 2023), allowing the generator to preserve source motion while leaving appearance and editing semantics to be determined by the edited anchors. The overall framework of the motion encoder is shown in Figure 2. More details about the architecture and training approach can be found in Appendix.

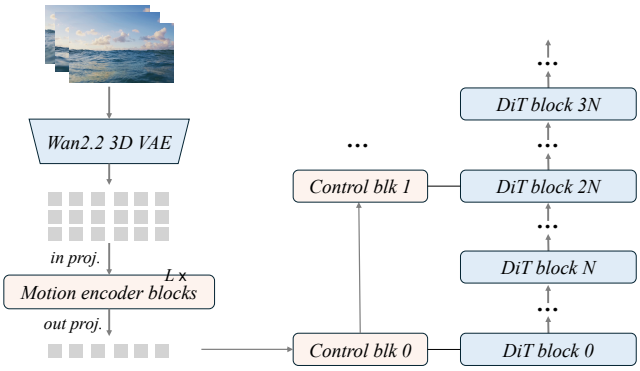

Figure 2: Architecture of the lightweight motion encoder used in Stage-2. The encoder projects video latents into a token sequence and applies factorized spatial-then-temporal self-attention to extract motion features that condition the frozen image-to-video backbone.

Anchor-preserving diffusion. Let D_ψ denote the pretrained Wan2.2 DiT. During generation, the edited keyframes are treated as *fixed anchors*: their latent features are clamped throughout the entire reverse diffusion process and are never denoised. Formally, we define a binary anchor mask

$$M_{\mathcal{A}} \in \{0,1\}^{T\times H\times W\times C}, \tag{2}$$

where $M_{\mathcal{A}}(t) = 1$ if frame $t \in \mathcal{A}$ and 0 otherwise. We encode the anchor frames by Wan2.2's 3D VAE encoder:

$$\hat{Z}_{\mathcal{A}} = \mathcal{V}_{\text{3D}}^{\text{enc}}(\hat{\mathbf{x}}_{\mathcal{A}}). \tag{3}$$

At the initial diffusion step S, the latent video is initialized as

$$X_S = M_{\mathcal{A}} \odot \hat{Z}_{\mathcal{A}} + (1 - M_{\mathcal{A}}) \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{4}$$

that is, anchor frames are directly set to the edited latent anchors, while all non-anchor frames start from random noise. At each reverse diffusion step s, the frozen DiT predicts the noise residual conditioned on the current latent video and the motion control:

$$\hat{\epsilon}_s = D_\psi(X_s, h_{\text{motion}}). \tag{5}$$

A standard diffusion update produces an intermediate estimate

$$\tilde{X}_{s-1} = \text{Step}(X_s, \hat{\epsilon}_s), \tag{6}$$

after which the anchor frames are re-imposed:

$$X_{s-1} = M_{\mathcal{A}} \odot \hat{Z}_{\mathcal{A}} + (1 - M_{\mathcal{A}}) \odot \tilde{X}_{s-1}. \tag{7}$$

Therefore, anchor features remain unchanged throughout denoising, while the model fills in all intermediate frames conditioned on both the fixed visual anchors and the source motion representation. After the final denoising step, the edited video is decoded back to pixel space by the 3D VAE decoder:

$$\hat{V} = \mathcal{V}_{\text{3D}}^{\text{dec}}(X_0). \tag{8}$$

4 Experiments

4.1 Experimental Setup

We use Wan2.2-I2V-A14B (Wan Team et al., 2025) as the video generation backbone. Unless otherwise specified, all pretrained components in Wan, including the 3D VAE encoder/decoder and the DiT generator, are kept frozen during training. The total number of learnable parameters is 710M, which is about 5% of the diffusion model's parameter count. In our experiments, we find that only lightweight training on the motion-related components is sufficient to achieve strong motion preservation and frame-to-video generation performance. This is mainly because we already effectively leverage the motion modeling and understanding capability of Wan2.2's pretrained 3D

Model	Total	Quality	Ins. Comp.	Fidelity	Motion	Sem. Consist.	Content
Ditto (Bai et al., 2026)	66.75	78.12	49.08	73.04	78.99	24.92	3.64
InsV2V (Cheng et al., 2024)	66.68	79.58	38.61	81.85	85.55	24.11	4.05
Lucy-Edit-Dev (DecartAI Team, 2025)	63.53	82.09	33.94	74.55	67.79	23.83	3.83
VACE (Jiang et al., 2025)	62.61	79.83	25.42	**82.58**	88.59	23.80	4.03
ICVE (Liao et al., 2025)	60.33	71.25	45.35	64.37	45.72	22.86	3.55
Omni-Video (Tan et al., 2025)	58.65	78.20	43.64	54.11	50.72	21.98	2.85
AnyV2V (Ku et al., 2024)	57.67	72.71	41.72	58.59	81.59	21.59	2.75
StableV2V (Liu et al., 2026)	50.89	69.17	42.67	40.84	75.13	19.78	1.79
AVE (ours)	**70.58**	**82.10**	**49.65**	80.01	**89.26**	**26.12**	**4.10**

Table 2: ***IVEBench short subset results.*** Here we report the overall performance of total score, video quality, instruction compliance, video fidelity, and detailed metrics of motion smoothness, semantic consistency, and content fidelity and leave the remaining scores of IVEBench in Appendix. Our AVE model consistently obtains the best results among the evaluation metrics.

Model	Total	Quality	Ins. Comp.	Fidelity	Motion	Sem. Consist.	Content
Ditto (Bai et al., 2026)	65.93	77.95	47.85	71.98	72.82	23.42	3.69
InsV2V (Cheng et al., 2024)	65.72	80.24	37.41	79.50	67.88	24.06	4.13
Lucy-Edit-Dev (DecartAI Team, 2025)	64.89	82.10	31.51	**81.05**	73.46	23.92	4.13
VACE (Jiang et al., 2025)	61.61	80.12	26.73	77.98	88.40	23.56	3.74
ICVE (Liao et al., 2025)	58.77	71.96	40.38	63.98	48.40	22.60	3.48
Omni-Video (Tan et al., 2025)	57.09	77.84	42.44	51.00	55.11	21.90	2.59
AnyV2V (Ku et al., 2024)	55.05	72.40	35.55	57.20	82.47	21.55	2.65
StableV2V (Liu et al., 2026)	50.93	69.37	42.09	41.33	77.33	20.41	1.79
AVE (ours)	**68.75**	**82.15**	**48.31**	75.79	**88.95**	**24.68**	**4.16**

Table 3: ***IVEBench long subset results.*** Similar to the short subset, our model attains the best results across the reported metrics. AVE has good robustness to long-video editing.

VAE and diffusion transformer. In practice, we train the motion encoder and ControlNet on only 500K video editing samples, and do not observe further improvements when scaling up either the data size or the training duration. We train our model on 64 A100 GPUs. We optimize the learnable parameters using AdamW with a base learning rate of 1×10^{-4} and weight decay 0.05. The optimizer uses $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We apply a cosine learning rate schedule with linear warmup at the beginning of training. All experiments are conducted in mixed precision. We evaluate our method on two public benchmarks, IVEBench (Chen et al., 2026b) and VIEBench (Mou et al., 2025), which assess instruction-guided video editing from different perspectives. Unless otherwise specified, we follow the standard evaluation protocols of each benchmark and report their official metrics.

4.2 Main Results

Tables 2 and 3 report the main results on the short and long subsets of IVEBench, respectively. AVE achieves the best performance on most reported metrics on both subsets, showing that our two-stage design leads to a strong and well-balanced video editing capability rather than improving only a single aspect of the task. On the short subset, AVE improves the total score from the previous best 66.75 to 70.58, while also establishing new best results in video quality (82.10), instruction compliance (49.65), and semantic consistency (26.12). The same trend holds on the long subset, where AVE reaches a total score of 68.75, clearly outperforming the previous best result. These results suggest that AVE does not merely optimize perceptual quality or instruction following; instead, it achieves a favorable balance among edit faithfulness, temporal coherence, and source-video preservation.

A key strength of AVE lies in motion-related quality. On IVEBench, AVE obtains the best motion score on both the short and long subsets, reaching 89.26 and 88.95, respectively. We attribute this advantage to the design of Stage-2. Instead of asking the video model to jointly solve semantic editing and temporal reasoning from scratch, AVE treats video generation as an anchor-constrained interpolation problem. The edited keyframes explicitly define the desired content at several time steps, while the Wan-based video generator fills in the intermediate frames under the guidance of a motion representation extracted from the source video. This division of labor makes the temporal generation process much more stable, and naturally reduces flickering, drift, and motion inconsistency

Method	Add	Swap / Change	Remove	Style / Tone	Ins. Follow	Pres.	Quality
Kling1.6 (Kling Team et al., 2025)	6.602	8.800	8.253	-	7.813	8.697	7.143
Kling-Omni (Kling Team et al., 2025)	**9.181**	9.194	9.133	**9.341**	9.518	9.346	**8.751**
Runway (Runway, 2025)	8.447	9.161	8.504	9.133	9.109	8.972	8.354
Pika (Pika, 2026)	-	7.408	-	-	7.542	7.847	6.837
MiniMax (Zi et al., 2025)	-	-	6.839	-	6.963	7.518	6.037
DiffuEraser (Li et al., 2025)	-	-	6.243	-	6.346	6.807	5.576
InsV2V (Cheng et al., 2024)	4.281	5.567	2.098	7.452	4.475	6.043	4.033
VACE (Jiang et al., 2025)	4.854	6.640	2.682	-	3.974	6.042	4.162
Omni-Video (Tan et al., 2025)	6.242	4.748	5.593	5.366	5.481	5.404	5.429
UniVideo (Wei et al., 2026)	8.656	8.683	8.233	8.711	8.708	8.963	8.042
InstructX (Mou et al., 2025)	8.349	9.072	8.322	9.196	9.059	8.905	8.241
SAMA (Zhang et al., 2026a)	8.711	9.340	9.144	9.259	9.344	9.370	8.626
AVE (ours)	9.054	**9.448**	**9.352**	9.203	**9.539**	**9.425**	8.630

Table 4: ***Comparison results on VIE-Bench.*** We report the average score of each editing category (add, swap, remove, and style) and each evaluation metric (instruction following, motion preservation, and video quality). Results of closed-source commercial models are marked in gray.

that commonly appear in naive per-frame editing and end-to-end video editing systems. In other words, AVE preserves motion well precisely because the video model is used for what it is best at: temporally coherent interpolation rather than full semantic editing.

Table 4 further shows that AVE generalizes well to VIE-Bench, which evaluates diverse editing types including add, swap/change, remove, and style/tone transfer. AVE is particularly strong on *swap/change* and *remove*, where it achieves the best average scores of 9.448 and 9.352, respectively. We believe these categories benefit the most from anchor-based generation, because the edited keyframes explicitly specify which object or region should be replaced or removed, and the video model only needs to propagate the edited result consistently through time. AVE also achieves the best averaged instruction-following score (9.539) and preservation score (9.425) across categories, indicating that it not only executes the edit faithfully, but also preserves the remaining content better than prior methods. This is exactly the behavior we would expect from our decomposition: Stage-1 handles the difficult semantic edit with a strong image editor, while Stage-2 focuses on preserving motion and completing the video around those edited anchors.

It is also worth noting that AVE remains competitive even against strong closed-source commercial systems. Although the best closed-source models still have a slight edge on a few appearance-centric categories such as *add* and *style/tone*, AVE already surpasses them on several important dimensions, including swap/change, remove, and the averaged instruction-following and preservation metrics. At the same time, AVE achieves a competitive overall quality score of 8.630, which is on par with or slightly better than the strongest open-source baselines. Taken together, these results show that AVE is not simply a lightweight approximation to end-to-end video editing. Rather, it provides a more effective formulation of the problem, enabling a video editor to directly inherit the strengths of modern image editing systems while retaining strong temporal coherence through a lightweight video generation stage.

Qualitative results. We provide qualitative examples in Fig. 3 to further illustrate the versatility of AVE across a wide range of editing scenarios. Our method performs well on diverse instruction types, including background removal, viewpoint change, object relocation, style transfer, local attribute editing, and explicit motion modification. A notable advantage of AVE is that, when the edit mainly changes appearance or scene content, the original motion and temporal structure are preserved very well. For example, in the first example, where the mountains are removed from the background, the rider's pose and motion remain nearly unchanged across frames, while the new background stays consistent and coherent over time. In the zoom-out example, AVE expands the room around the father and daughter while maintaining stable identity, posture, and scene layout. The beach example further shows that AVE can handle object relocation reliably: the woman is moved to the right side of the frame, while the coastline and ocean remain temporally coherent.

The second row demonstrates that AVE is also effective on more challenging edits. In the watercolor example, the style is transformed consistently across frames while preserving the wave structure and overall scene composition, showing that AVE can handle global appearance changes rather than



Figure 3: ***Instruction-based video editing results produced by AVE.*** Guided by keyframe-level instructions and image-to-video extension, our method delivers high-fidelity visual details, strong temporal coherence, and robust performance across diverse editing tasks.

only local edits. In the tattoo removal example, the model performs a subtle local modification while keeping the surrounding body shape and background stable, indicating strong fine-grained control. Most importantly, the final example shows that AVE can support substantial motion changes: the couple is edited to start spinning while embracing, yet the generated motion remains smooth and stable instead of becoming abrupt or jittery. Together, these examples highlight the core strength of AVE: strong semantic edits are established at the keyframe level, and the motion-guided interpolation stage then propagates them through time in a controlled and coherent manner.

4.3 Ablation Study

Unless otherwise specified, all ablations are conducted on the long subset of IVEBench. Each experiment is done by three runs and the standard deviation is less than 0.1%. We focus on two questions: (1) how much the final video editing performance depends on the image editor used in Stage-1, and (2) how sensitive AVE is to the keyframe selection strategy.

Effect of the Stage-1 image editor. Table 5 studies the impact of replacing the image editor in Stage-1 while keeping Stage-2 unchanged. By default, AVE uses Nano Banana 2, which yields the best overall results. Replacing it with other strong closed-source editors such as GPT-5.1 or Seedream 4.0 only causes a mild drop, while switching to our finetuned BAGEL leads to a further decrease. Using the original untuned BAGEL causes a much larger degradation. This trend provides a particularly important insight: the editing ability of AVE is strongly correlated with the quality of the image editor. Notably, the largest drops appear in instruction compliance, semantic consistency, and content fidelity, while motion remains relatively stable. This is consistent with our design: motion is mainly handled by Stage-2, which is fixed across all variants, whereas the semantic correctness of the edit is largely determined by the anchor frames produced by Stage-1. In other words, better image editing almost directly translates into better video editing. This strongly supports our main claim that the more difficult part of video editing is still the underlying image editing problem, while the image-to-video transfer itself is comparatively simple once high-quality anchors are available.

It is also worth noting that the tuned BAGEL model significantly outperforms the original untuned BAGEL. This suggests that multi-image consistent editing is important even before the video generation stage, and validates our choice of adapting open-source image editors with multi-frame video-derived supervision when closed-source APIs are unavailable.

Stage-1 Image Editor	Total	Quality	Ins. Comp.	Fidelity	Motion	Sem. Consist.	Content
Nano Banana 2 (default)	**68.75**	**82.15**	**48.31**	**75.79**	**88.95**	**24.68**	**4.16**
GPT-5.1	68.10	81.43	47.11	75.76	88.74	24.10	4.02
Seedream 4.0	67.59	80.36	46.67	75.74	88.61	23.83	3.85
BAGEL (finetuned)	66.07	79.38	43.06	75.78	88.10	21.96	3.83
BAGEL (base)	58.54	68.12	37.84	69.66	77.21	20.34	3.42

Table 5: ***Ablation on the Stage-1 image editor on IVEBench-long.*** We replace only the composed image editor in Stage-1 and keep the entire Stage-2 pipeline fixed. The results show a strong correlation between image editing quality and final video editing performance.

Strategy	#Keyframes	Total	Quality	Ins. Comp.	Fidelity	Motion
Uniform	2	66.85	79.58	46.02	74.96	87.42
Uniform	$\lfloor T/48 \rfloor$	67.73	80.84	47.12	75.24	88.21
Uniform	$\lfloor T/24 \rfloor$	68.21	81.69	47.86	75.09	88.71
Uniform	$\lfloor T/12 \rfloor$	67.94	81.27	47.40	75.15	88.38
Scene-first + max-gap (ours)	2	67.28	79.93	46.48	75.44	87.67
Scene-first + max-gap (ours)	$\lfloor T/48 \rfloor$	68.09	81.16	47.56	75.55	88.46
Scene-first + max-gap (ours)	$\lfloor T/24 \rfloor$	**68.75**	**82.15**	**48.31**	**75.79**	**88.95**
Scene-first + max-gap (ours)	$\lfloor T/12 \rfloor$	68.34	81.61	47.83	75.59	88.54

Table 6: ***Ablation on keyframe selection strategies and keyframe budgets on IVEBench-long.*** Our default strategy combines scene-aware anchors from PySceneDetect with largest-gap completion. AVE is robust to the exact keyframe selection rule, while a moderate keyframe budget around $\lfloor T/24 \rfloor$ gives the best trade-off.

Effect of keyframe selection. Table 6 compares different keyframe selection strategies and budgets. Our default strategy first extracts scene-aware anchors using PySceneDetect, keeps the first frame of each scene together with the final frame of the video, and then fills the remaining budget using a largest-gap rule. We compare this strategy against simple uniform sampling under different keyframe budgets. Two observations are clear. First, the gap between the two strategies is relatively small across all settings, indicating that AVE is fairly robust to the exact keyframe selection rule. This is encouraging, as it suggests that once a sparse set of reasonable anchors is provided, the Stage-2 interpolation model can reliably propagate the edits to the full video. Still, our scene-aware strategy is consistently better than uniform sampling, especially when the budget is limited, likely because it is more likely to place anchors near semantic transitions and scene changes.

Second, the best performance is achieved when the number of keyframes is set to approximately $\lfloor T/24 \rfloor$. Using too few keyframes leads to weaker instruction compliance and lower fidelity because the anchors become too sparse to fully specify the desired edit over long sequences. On the other hand, using too many keyframes also hurts performance slightly. We hypothesize that this is because the Wan2.2 3D VAE already performs a $4\times$ temporal downsampling, so overly dense anchors reduce the effective interpolation space and make temporal completion less stable. As a result, AVE performs best when the anchors are sparse enough to allow interpolation, but dense enough to stabilize the desired edit.

5 Conclusion

In this paper, we propose Anchor-based Video Editing (AVE), a simple two-stage framework that transforms image editors into video editors. Our key insight is that the main difficulty of video editing still lies in image-level semantic editing, while temporal propagation can be handled much more efficiently once a sparse set of high-quality edited keyframes is available. Based on this observation, AVE first performs composed image editing on selected keyframes, and then uses a motion-guided image-to-video diffusion model to interpolate the full edited video. This design directly inherits the strengths of modern image editing systems while avoiding expensive end-to-end video editing training. Experiments on IVEBench and VIE-Bench show that AVE achieves strong performance across instruction following, semantic consistency, content fidelity, and motion preservation, and our ablations further reveal that final video editing quality is strongly correlated with the quality of the

image editor. We hope this work offers useful insights into building future video editing systems in a more modular, efficient, and scalable way.

Acknowledgments

Feng Wang and Alan Yuille are funded by ONR N000142412696.

References

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. In *CVPR*, 2026.

Black Forest Labs. Introducing FLUX.1 Kontext and the BFL playground. https://bfl.ai/blog/flux-1-kontext, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. VideoCrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024.

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

Yinan Chen, Jiangning Zhang, Teng Hu, Yuxiang Zeng, Zhucun Xue, Qingdong He, Chengjie Wang, Yong Liu, Xiaobin Hu, and Shuicheng Yan. IVEBench: Modern benchmark suite for instruction-guided video editing assessment. In *ICLR*, 2026b.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. In *ICLR*, 2024.

DecartAI Team. Lucy Edit: Open-weight text-guided video editing. https://huggingface.co/decart-ai/Lucy-Edit-Dev, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Alisa Fortin and Bea Alessio. Build with Nano Banana 2, our best image generation and editing model. https://blog.google/innovation-and-ai/technology/developers-tools/build-with-nano-banana-2/, 2026.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024.

Google. Gemini 3. https://gemini.google.com/, 2025.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024.

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. In *ECCV*, 2026.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, and Ying Shan. SmartEdit: Exploring complex instruction-based image editing with multimodal large language models. In *CVPR*, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *ICCV*, 2025.

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

Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research*, 2024.

Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. DiffuEraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025.

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.

Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. StableV2V: Stabilizing shape consistency in video-to-video editing. *IEEE Transactions on Circuits and Systems for Video Technology*, 36(4):5467–5482, 2026.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1X-Edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Chong Mou, Qichao Sun, Yanze Wu, Pengze Zhang, Xinghui Li, Fulong Ye, Songtao Zhao, and Qian He. InstructX: Towards unified visual editing with MLLM guidance. *arXiv preprint arXiv:2510.08485*, 2025.

OpenAI. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/, 2025.

Pika. Pika. https://pika.art/, 2026.

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 *ICCV*, 2023.

Qwen Team. Qwen-Image-Edit: Image editing with higher quality and efficiency. https://qwen.ai/blog?id=qwen-image-edit, 2025.

Zhefan Rao, Bin Zou, Haoxuan Che, Xuanhua He, Chong Hou Choi, Yanheng Li, Rui Liu, and Qifeng Chen. InsEdit: Towards instruction-based visual editing via data-efficient video diffusion models adaptation. *arXiv preprint arXiv:2604.08646*, 2026.

Runway. Introducing Runway Gen-4. https://runwayml.com/research/introducing-runway-gen-4, 2025.

Cuifeng Shen, Yulu Gan, Chen Chen, Xiongwei Zhu, Lele Cheng, Tingting Gao, and Jinzhi Wang. Decouple content and motion for conditional image-to-video generation. In *AAAI*, 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.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Wan Team, 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.

Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. InstructEdit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*, 2023a.

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional video synthesis with motion controllability. In *NeurIPS*, 2023b.

Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. UniVideo: Unified understanding, generation, and editing for videos. In *ICLR*, 2026.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, 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 *ICCV*, 2023.

Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. InsViE-1M: Effective instruction-based video editing with elaborate dataset construction. In *ICCV*, 2025.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. OmniGen: Unified image generation. In *CVPR*, 2025.

Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. InstructVideo: Instructing video diffusion models with human feedback. In *CVPR*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

Xinyao Zhang, Wenkai Dong, Yuxin Song, Bo Fang, Qi Zhang, Jing Wang, Fan Chen, Hui Zhang, Haocheng Feng, Yu Lu, et al. SAMA: Factorized semantic anchoring and motion alignment for instruction-guided video editing. *arXiv preprint arXiv:2603.19228*, 2026a.

Zhongwei Zhang, Fuchen Long, Wei Li, Zhaofan Qiu, Wu Liu, Ting Yao, and Tao Mei. In-context generation with regional constraints for instructional video editing. In *ICML*, 2026b.

Bojia Zi, Weixuan Peng, Xianbiao Qi, Jianan Wang, Shihao Zhao, Rong Xiao, and Kam-Fai Wong. MiniMax-Remover: Taming bad noise helps video object removal. In *NeurIPS*, 2025.

Appendix

Formal Details of The Motion Encoder

Let $U^{(\ell-1)}$ denote the input of the ℓ-th layer. We first perform spatial attention independently within each frame:

$$\tilde{U}^{(\ell)}_{t,:,:} = U^{(\ell-1)}_{t,:,:} + \mathrm{Attn}_{\text{spatial}}\Big(\mathrm{LN}(U^{(\ell-1)}_{t,:,:})\Big), \tag{9}$$

where $\mathrm{Attn}_{\text{spatial}}$ attends over all spatial locations (h, w) at fixed time t. We then perform temporal attention independently at each spatial location:

$$\bar{U}^{(\ell)}_{:,h,w} = \tilde{U}^{(\ell)}_{:,h,w} + \mathrm{Attn}_{\text{temporal}}\Big(\mathrm{LN}(\tilde{U}^{(\ell)}_{:,h,w})\Big), \tag{10}$$

where $\mathrm{Attn}_{\text{temporal}}$ attends over all timesteps at fixed spatial position (h, w). Finally, the output of the layer is obtained with a feed-forward network:

$$U^{(\ell)} = \bar{U}^{(\ell)} + \mathrm{MLP}\Big(\mathrm{LN}(\bar{U}^{(\ell)})\Big). \tag{11}$$

After $L_m = 6$ layers, the final motion representation is

$$h_{\text{mot}} = U^{(L_m)}. \tag{12}$$

We inject h_{mot} into the frozen Wan2.2 DiT through a ControlNet-style residual branch (Zhang et al., 2023), allowing the generator to preserve source motion while leaving appearance and editing semantics to be determined by the edited anchors. The overall framework of the motion encoder is shown in Figure 2.

Training Stage-2

To train the lightweight motion branch, we keep the Wan 3D VAE and DiT frozen and optimize only $\mathcal{M}_\theta$ and the ControlNet-style residual branch. Given a training target video $V^\star$ with latent code

$$Z^\star = \mathcal{V}^{\text{enc}}_{\text{3D}}(V^\star), \tag{13}$$

we sample a diffusion timestep τ and noise ϵ, and obtain a noisy latent video

$$X^\star_\tau = \alpha_\tau Z^\star + \sigma_\tau \epsilon. \tag{14}$$

To mimic inference, the selected anchor frames are replaced by their clean latent values:

$$\tilde{X}^\star_\tau = M_{\mathcal{A}} \odot Z^\star_{\mathcal{A}} + (1 - M_{\mathcal{A}}) \odot X^\star_\tau. \tag{15}$$

The model predicts the noise residual

$$\hat{\epsilon} = D_\psi(\tilde{X}^\star_\tau, h_{\text{mot}}), \tag{16}$$

and we compute the training loss only on non-anchor frames:

$$\mathcal{L}_{\text{vid}} = \|(1 - M_{\mathcal{A}}) \odot (\hat{\epsilon} - \epsilon)\|_2^2. \tag{17}$$

This objective explicitly teaches the model to interpolate and propagate content between fixed anchors, rather than re-synthesizing the anchors themselves.

Broader Impacts

This work focuses on foundational research in video generation. We did not identify any specific direct societal impacts beyond the general considerations that accompany generative media.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims reflect the contributions and scope

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Method and Experiments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Training detail is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available at https://github.com/wangf3014/AVE.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: They are provided in Section 3 and 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Reported in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Reported in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: No ethics issues.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussed in Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets are mentioned.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: It's documented.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: N/A.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are used in some benchmarks for evaluation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.