MAKING MLLM LATENTS MATTER: FACTOR-AWARE PREDICTIVE LEARNING FOR CINEMATIC VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cinematic video generation requires both temporal inference from limited inputs (*i.e.*, sparse rather than frame-by-frame guidance) and factor-aware control of character, scene, action, and camera. Multimodal large language models (MLLMs) offer useful priors for both capabilities, but their representations are not automatically actionable for video generation: a Video Diffusion Transformer (DiT) may bypass them through native conditioning pathways, and using them does not ensure either factor-aware control or generation-compatible temporal prediction. We propose **Factor-Aware Predictive Learning (FAPL)**, which establishes effective MLLM latent use as a shared prerequisite and then targets the two capabilities with complementary learning signals. **First, latent-only generator grounding** trains the DiT with a frozen MLLM and connector, strengthening the DiT's use of MLLM latents for generation. **Second, for factor-aware control**, paired videos specify a target-factor change and non-target preservation, and factor-span replacement trains the DiT to realize this relationship in its output. **Third, for temporal inference**, text-only and text-plus-complete-video observations share a video target, and the trained DiT is then frozen while the MLLM and connector are optimized through it. This final stage trains limited-input representations for generation while retaining complete-observation reconstruction as a complementary training mode. Extensive experiments demonstrate our consistently better performance on three cinematic video generation tasks (*i.e.*, sparse-to-video, factor-aware editing, and multi-shot generation) over baselines and validate FAPL's contributions to MLLM feature use, temporal control, and factor coordination.

1 INTRODUCTION

Cinematic video generation (Wang et al., 2025a; Wu et al., 2026; Zhang et al., 2026b) helps creators explore content, performances, and camera choices before physical production. Beyond visual realism, it requires two capabilities (Fig. 1). **(i) Temporal inference from limited inputs** requires the model to construct plausible action sequences and camera motion from a script and a few visual references (Xiao et al., 2026; Zhang et al., 2026a), which only sparsely specify how a shot unfolds (*i.e.*, they do not provide a dense, frame-by-frame specification of all visual content). **(ii) Factor-aware control** requires character, scene, action, and camera to follow their respective instructions while remaining compatible: changing a character should preserve the requested action, and changing camera motion should not replace the scene (Huang et al., 2026b; Girish et al., 2026). Therefore, the central challenge is not only to infer missing content, but also to make those inferences selectively controllable in the generated video.

Standard conditioning provides limited support for these requirements. Text encoders (Raffel et al., 2020; Radford et al., 2021) offer limited visual and temporal grounding, while VAE-encoded references (Kingma & Welling, 2013) preserve appearance without explicitly representing action or camera intent. Multimodal large language models (MLLMs) (Yang et al., 2025; Wang et al., 2024) provide a promising alternative (Wang et al., 2026): their visual priors and semantic reasoning can help interpret scripts and references, infer plausible temporal details, and coordinate cinematic factors. Recent methods (Yang et al., 2026a; Wei et al., 2026) therefore condition Video Diffusion Transformers (DiTs) (Peebles & Xie, 2023) on projected MLLM hidden states.

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

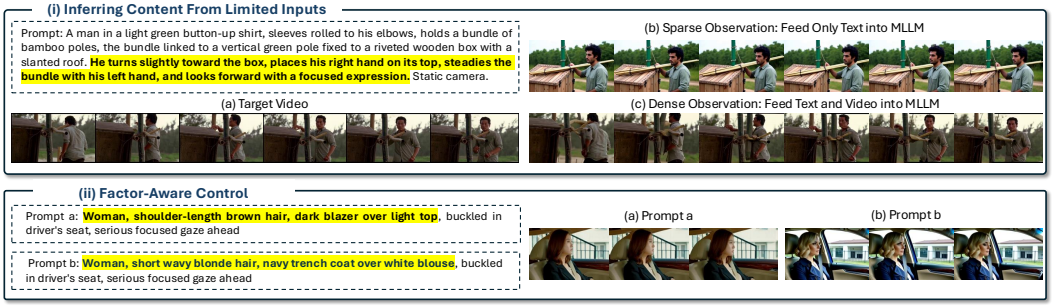

Figure 1: **Cinematic video generation requires inference from limited inputs and factor-aware control.** (i) The man turns toward the box, places his right hand on top, and steadies the bundle with his left hand. Text-only conditioning misses this sequence, while observing the complete video enables the same model to reproduce it. (ii) Keeping the descriptions of all other factors fixed and replacing only the character-related prompt, the non-character factors change as well. *These cases illustrate failures in temporal inference and factor control, with editing providing a probe of what the generator changes and preserves.* Note that prompts in the main paper are abbreviated or keyword-level, and full prompts and videos are given in the supplementary HTML files.

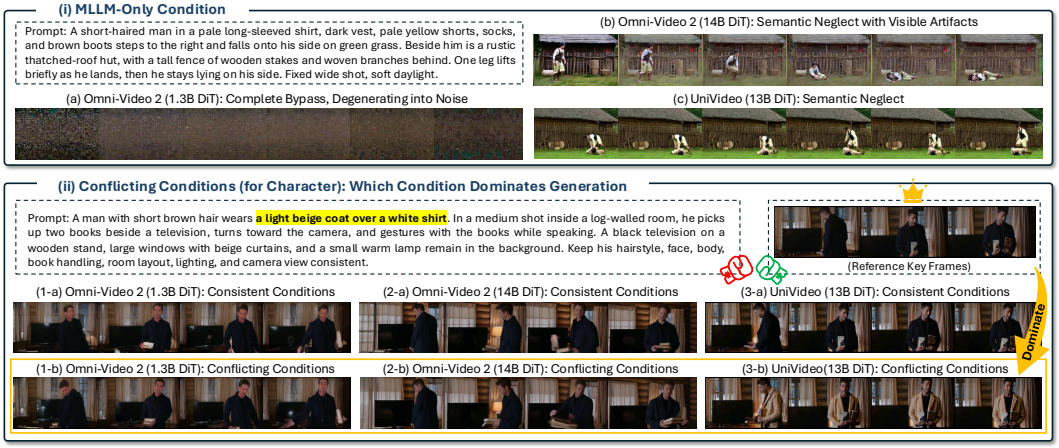

Figure 2: **MLLM latents are easily bypassed in existing MLLM–DiT models.** (i) Under MLLM-only conditioning, Omni-Video 2 (1.3B DiT) produces noise, while Omni-Video 2 (14B DiT) and UniVideo produce artifacts and depict one person's successive actions as two people acting differently. MLLM latents alone therefore fail to provide reliable generation guidance in these examples. (ii) Under conflicting conditions, VAE references dominate the output.

However, making MLLM features available is different from making them effective for generation. Two obstacles remain: **First, MLLM feature bypass** is a shared bottleneck for both capabilities: native text-encoder and VAE-reference pathways can reduce the generation loss without substantial reliance on MLLM latents (Fig. 2). **Second, task-structured supervision is insufficient**. For factor-aware control, ordinary video-level training does not explicitly associate an intervention on a particular MLLM span with the intended output change and non-target preservation. For temporal inference, adapting the connector and DiT around a frozen MLLM does not train the MLLM itself to produce temporally informative representations from sparse inputs. Thus, the generator must learn how to use MLLM representations, while the MLLM must also learn what representations to produce for generation.

We propose **Factor-Aware Predictive Learning (FAPL)** to strengthen the use of MLLM representations for temporal inference and factor-aware control through three training stages. **First, to strengthen the DiT's use of MLLM features**, Stage I freezes the MLLM and connector and trains the DiT with their projected latents as the sole content conditions, using both text-only and text-plus-complete-video observations. This establishes a generation interface that subsequent stages can use for factor-aware control and predictive learning. **Second, to enable factor-aware control**, Stage II uses paired videos that change selective cinematic factors while preserving the others. Structured descriptions locate the projected latent spans corresponding to character, scene, action, and camera.

Keeping the MLLM and connector frozen, we replace the selected factor span between paired examples and train the DiT against the corresponding video target. The paired videos specify what should change and remain, while span replacement explicitly associates this visual relationship with a localized conditioning intervention, without assuming fully disentangled representations. **Finally, to learn temporal inference from sparse inputs**, Stage III freezes the trained DiT and jointly optimizes the MLLM and connector using text-only and text-plus-complete-video observations that share a video target. The text-only mode trains prediction from limited evidence, while the complete-video mode provides complementary reconstruction supervision through the same generation interface. The fixed DiT directs learning toward generation-compatible representations without simultaneous decoder adaptation. Experiments evaluate FAPL on sparse-to-video generation, factor-aware editing, and multi-shot generation, with stage-wise ablations examining the respective contributions of strengthened MLLM feature use, factor interventions, and mixed-observation predictive learning.

Our contributions are threefold: (i) We investigate the gap between multimodal understanding and its effective use in cinematic generation, and propose **FAPL** to translate MLLM understanding into actionable guidance for inference from limited inputs and factor-aware control. (ii) We introduce factor-aware supervision through latent interventions, turning paired edits into structured learning signals for how the generator should change and preserve cinematic content, without assuming fully disentangled representations. (iii) We formulate predictive learning through a fixed generator, linking limited-input prediction and complete-observation reconstruction via shared video targets to learn temporally informative, generation-compatible MLLM representations.

2 Related Work

Cinematic Video Generation. Cinematic video generation (Yang et al., 2026c; Elmoghany et al., 2025) has recently received growing attention for producing shots with specified content, performances, and camera motions from limited materials (Wang et al., 2025a; Meng et al., 2026; Yang et al., 2026b). Existing studies target the two requirements: (i) Inferring content from limited inputs. Reference-driven and multi-shot methods maintain identity and scene consistency from given references (Xiao et al., 2026; Zhang et al., 2026a; Liu et al., 2025). (ii) Factor-Aware Control. Camera motion, character replacement, action transfer, and scene modification are typically handled separately (Wang et al., 2023; Tan et al., 2026a; Ye et al., 2026; Jiang et al., 2025; Huang et al., 2026b; Girish et al., 2026; Fang et al., 2026; Huang et al., 2026a).

MLLM-Conditioned Video Generation and Editing. Recent video diffusion models achieve high-quality synthesis and support a growing range of editing tasks (Blattmann et al., 2023; Kong et al., 2024; Wan et al., 2025). To move beyond single-task systems, recent work couples MLLMs with diffusion backbones, either through in-context conditioning (Ye et al., 2026; Ju et al., 2025; Jiang et al., 2025) or through an MLLM understanding branch connected to a generation branch (Wei et al., 2026; Yang et al., 2026a; Tan et al., 2026b). Related studies improve controllability over specific factors such as camera motion (Wang et al., 2023) and subject identity (Liu et al., 2025). These systems differ in architecture and task coverage, but commonly inject MLLM features alongside text-encoder and VAE reference features through a shared conditioning interface, and supervise training at the video level. Such supervision does not determine how much each condition contributes, and the generator may rely on familiar text and dense visual-reference pathways. Paired editing examples are also typically consumed as additional supervised samples, without explicitly indicating which factor should change and which content should be preserved. *In contrast, FAPL strengthens the role of MLLM latents through latent-only training, and uses factor-span replacement between paired examples to supervise factor-aware changes and preservation.*

Predictive Representation Learning. Joint-embedding predictive architectures learn representations by predicting target features from partial observations rather than reconstructing pixels, avoiding unpredictable low-level details (Assran et al., 2023), and extend to action-conditioned future prediction for planning and control (Bardes et al., 2024; Assran et al., 2025). Generative models also decode predicted representations into visual outputs. However, such models usually learn task-specific spaces without high-fidelity video decoders, or require separately trained decoders, and rarely use paired editing data to supervise factor-level change and preservation. *In contrast, we reuse a pretrained MLLM and Video DiT and supervise prediction through visual reconstruction with a fixed generator, preserving the decoding pathway while training the MLLM to infer video content from limited observations.*

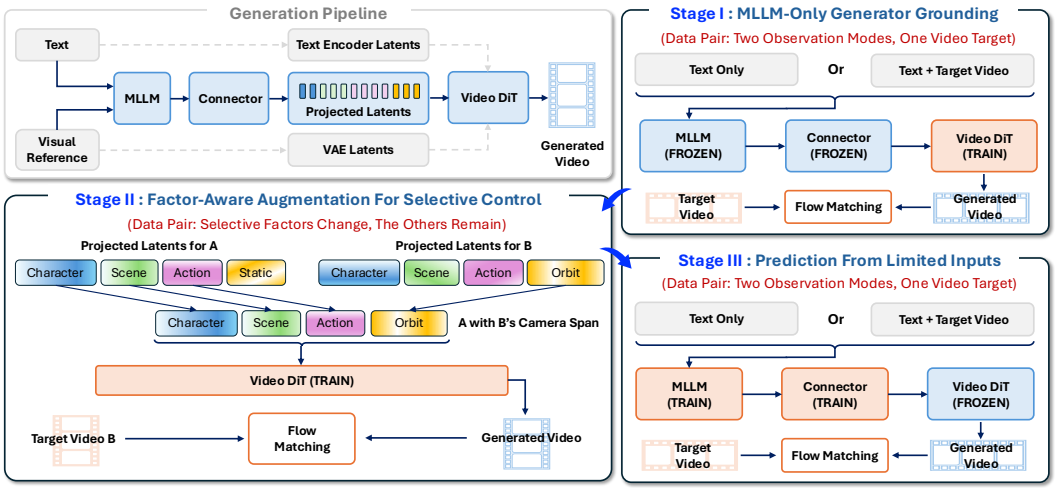

Figure 3: **Overview of Factor-Aware Predictive Learning (FAPL).** Stage I freezes the MLLM and its original connector and trains the DiT using MLLM latents. Stage II retains this configuration and swaps factor-specific output spans between paired examples to supervise factor-aware changes. Stage III freezes the trained DiT and jointly updates the MLLM and connector through video reconstruction, and text-only and text-plus-complete-video inputs share the same generation target.

3 METHOD

Factor-Aware Predictive Learning (FAPL) trains MLLM representations for limited-input temporal inference and factor-aware control of character, scene, action, and camera. It structures supervision in generation space: **factor-span interventions** use paired edits to link localized latent changes to target-factor changes and non-target preservation, while **shared-target predictive learning** couples limited-input prediction with complete-observation reconstruction through a fixed generator. MLLM-only conditioning strengthens the DiT's use of these representations, supporting both mechanisms without architectural changes.

We first describe the framework and paired data in Sec. 3.1. Training then proceeds in three stages: (i) MLLM-Only Generator Grounding strengthens the DiT's use of MLLM features (Sec. 3.2); (ii) Factor-Aware Control Improvement trains the DiT to realize target-factor changes while preserving other content through factor-span replacement (Sec. 3.3); and (iii) Prediction From Limited Inputs optimizes the MLLM and connector through the fixed DiT using shared targets across observation modes (Sec. 3.4). All three stages use MLLM latents as the only content conditions. Fig. 3 illustrates the framework, Algorithm 1 summarizes training, and Sec. 3.5 describes inference.

3.1 PRELIMINARY: FRAMEWORK AND PAIRED DATA

The generator (*i.e.*, Video DiT) must learn both to use MLLM information and to respond appropriately in a factor-aware manner. We provide these signals through paired observations and paired edits, without introducing a new representation network (which we discussed in the Appendix B).

3.1.1 FRAMEWORK: MLLM-CONDITIONED VIDEO DIFFUSION

MLLM-conditioned video diffusion (Yang et al., 2026a; Wei et al., 2026) connects an MLLM to a pretrained Video DiT. The DiT is trained through flow matching (Lipman et al., 2022) and receives three concatenated conditions: $C = [C^{\text{text}}; C^{\text{mllm}}; C^{\text{ref}}]$, comprising text-encoder features, MLLM features, and VAE-encoded visual features, respectively. Here, $[\cdot;\cdot]$ denotes concatenation along the token dimension. Individual backbones may omit a pathway. For example, UniVideo (Wei et al., 2026) removes the original text encoders.

The MLLM condition is produced through the connector. Given a text or multimodal input u, the MLLM F_θ produces hidden states $H_\theta(u) = F_\theta(u)$, which the connector P_η maps to the DiT conditioning space:

$$C^{\text{mllm}} = Z_{\theta,\eta}(u) = P_\eta\big(H_\theta(u)\big), \tag{1}$$

Algorithm 1 Three-Stage Training of Factor-Aware Predictive Learning (FAPL)

Require: Pretrained MLLM F_{θ_0}, connector P_{η_0}, and DiT G_{ϕ_0}; observation data $\mathcal{D}_{\text{obs}}$ and editing pairs $\mathcal{D}_{\text{edit}}$.

Initialize $\theta \leftarrow \theta_0$, $\eta \leftarrow \eta_0$, and $\phi \leftarrow \phi_0$; Use only MLLM latents as content conditions in all stages.
Stage I: Enhance MLLM Latent Use (MLLM-Only Generator Grounding).
for each training step **do**
  Sample $(x, p) \sim \mathcal{D}_{\text{obs}}$ and observation mode $m \sim \rho$; Compute $Z_{\theta_0,\eta_0}(u^m)$.
  Update ϕ using $\mathcal{L}_{\text{FM}}(x \mid Z_{\theta_0,\eta_0}(u^m); \phi)$; keep θ_0, η_0 fixed.
end for
Stage II: Improve Factor-Aware Control.
for each training step **do**
  Sample (x^a, p^a, x^b, p^b, k) from $\mathcal{D}_{\text{edit}}$; Compute Z^a, Z^b and locate their factor-k spans.
  Replace the designated spans as in Eq. 4; Update ϕ using $\mathcal{L}_{\text{edit}}$; keep θ_0, η_0 fixed.
end for
Save $\phi^* \leftarrow \phi$ and freeze the DiT.
Stage III: Learn Prediction From Limited Inputs.
for each training step **do**
  Sample (x, p) from $\mathcal{D}_{\text{obs}}$ and observation mode $m \sim \pi$.
  Construct $u^{\text{t}} = p$ if $m = \text{t}$, and $u^{\text{tv}} = (p, x)$ otherwise.
  Compute $Z_{\theta,\eta}(u^m)$ and update θ, η using $\mathcal{L}_{\text{FM}}(x \mid Z_{\theta,\eta}(u^m); \phi^*)$; keep ϕ^* fixed.
end for

where θ and η are the MLLM and connector parameters. We retain the original hidden-state extraction and connector. Our factor-level interventions operate on the projected sequence $Z_{\theta,\eta}(u)$, after the connector and before the DiT.

For a target video x with VAE latent z_0, flow matching constructs $z_t = \alpha_t z_0 + \sigma_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise, t is the flow time, and α_t, σ_t specify the interpolation schedule. The DiT G_ϕ, with parameters ϕ, predicts the target velocity $v^* = \partial z_t / \partial t$:

$$\mathcal{L}_{\text{FM}}(x \mid C; \phi) = \mathbb{E}_{t,\epsilon}\left[\left\|G_\phi(z_t, t \mid C) - v^*\right\|_2^2\right]. \tag{2}$$

We follow the pretrained backbone's flow schedule and sampling procedure. This objective supervises generation under the combined conditions, but does not determine their individual contributions. The DiT may rely primarily on text and VAE features, so accurate reconstruction alone does not establish effective use of MLLM latents.

3.1.2 Paired Data

Two Observation Modes, One Video Target. For each training video x and its description p, we construct two inputs: $u^{\text{t}} = p$ and $u^{\text{tv}} = (p, x)$. Both are supervised with the same target video x. The text-only input requires inference from a description, while the text-plus-video input provides direct visual evidence. We denote these examples by $\mathcal{D}_{\text{obs}}$. The complete target video is never supplied to the text-only branch.

Selective Factors Change, the Others Remain. We construct the editing set $\mathcal{D}_{\text{edit}}$ through factor-specific selection and generation of source–target video pairs, covering changes to character, scene, action, and camera motion. Each tuple (x^a, p^a, x^b, p^b, k) contains two videos x^a, x^b and their descriptions p^a, p^b, with only the selected factor $k \in \mathcal{F} = \{\text{character}, \text{scene}, \text{action}, \text{camera}\}$ modified. Both descriptions are structured prompts: we rewrite each caption into four ordered fields, character, scene, action, and camera, and obtain p^b by updating only the edited factor's field while retaining the others. Field labels and boundaries locate the corresponding token spans in the MLLM hidden states and projected latents, giving each factor an explicit address for intervention. Span lengths follow the text and need not match across examples, and we do not assume the contextualized spans are fully disentangled.

3.2 Stage I: MLLM-Only Generator Grounding

MLLM representations can influence generation only if the DiT can reliably decode and use them. In the original architecture, native text-encoder and VAE-reference pathways allow the DiT to reduce the generation loss without sufficiently relying on MLLM latents. Stage I therefore isolates the MLLM pathway and trains the DiT under both limited and complete observations. We freeze the pretrained MLLM and connector and update only the DiT.

Mixed-Observation Latent-Only Grounding. For each video x and description p, we use the text-only input $u^{\mathrm{t}} = p$ and the text-plus-complete-video input $u^{\mathrm{tv}} = (p, x)$. The frozen MLLM and connector produce $Z_{\theta_0,\eta_0}(u^m)$ for observation mode $m \in \{\mathrm{t}, \mathrm{tv}\}$. In both modes, the DiT receives MLLM latents as its only content condition; the separate text-encoder and VAE-reference conditions are disabled. Dense observations ground the DiT in the visual and temporal information carried by MLLM latents, while text-only observations maintain compatibility with limited-input generation and the text-based interventions in Stage II.

Mixed-Observation Objective. Stage I minimizes

$$\mathcal{L}_{\mathrm{ground}}(\phi) = \mathbb{E}_{(x,p)\sim\mathcal{D}_{\mathrm{obs}}}\left[\sum_{m\in\{\mathrm{t},\mathrm{tv}\}} \rho_m \, \mathcal{L}_{\mathrm{FM}}\left(x \mid Z_{\theta_0,\eta_0}(u^m); \phi\right)\right], \tag{3}$$

where $\rho_{\mathrm{t}} + \rho_{\mathrm{tv}} = 1$. We optimize ϕ while keeping θ_0 and η_0 fixed. Because alternative content pathways are unavailable, the DiT must learn to use MLLM latents across both observation modes. Stage I adapts the generator to fixed MLLM representations, while Stage III later reverses this direction and adapts the MLLM pathway through a fixed generator.

3.3 Stage II: Factor-Aware Control Improvement

Using MLLM features is not sufficient for factor-aware control. The DiT must also learn which output changes correspond to a particular change in its conditions. Stage II introduces factor-level latent augmentation with explicit video targets. The MLLM and connector remain frozen, and training uses text-only inputs with MLLM-latent-only conditioning.

Factor-Span Replacement. For a pair in $\mathcal{D}_{\mathrm{edit}}$, let $Z^a = Z_{\theta_0,\eta_0}(p^a)$ and $Z^b = Z_{\theta_0,\eta_0}(p^b)$ be the projected latents of the two descriptions. Define $\mathcal{R}_k(A, B)$ as replacing the factor-k span in A with the corresponding span from B, leaving the remaining tokens unchanged. The augmented conditions are

$$Z_k^{a\leftarrow b} = \mathcal{R}_k(Z^a, Z^b), \qquad Z_k^{b\leftarrow a} = \mathcal{R}_k(Z^b, Z^a). \tag{4}$$

Replacement is applied after the connector and before the DiT, so no additional projection follows. Complete spans are removed and inserted, so replacement does not require equal token counts; sequence lengths and masks are updated accordingly. Since the pair changes only factor k, the condition $Z_k^{a\leftarrow b}$ is supervised with x^b, and the reverse condition is supervised with x^a.

Editing Objective. We update the DiT using both replacement directions:

$$\mathcal{L}_{\mathrm{edit}}(\phi) = \mathbb{E}_{(x^a,p^a,x^b,p^b,k)\sim\mathcal{D}_{\mathrm{edit}}}\left[\mathcal{L}_{\mathrm{FM}}\big(x^b \mid Z_k^{a\leftarrow b}; \phi\big) + \mathcal{L}_{\mathrm{FM}}\big(x^a \mid Z_k^{b\leftarrow a}; \phi\big)\right]. \tag{5}$$

The target video supervises the effect of the replaced span and preservation of the remaining factors. Thus, factor awareness is learned through the generator's response to controlled replacements, rather than through a feature-separation objective. We denote the DiT parameters after this stage by ϕ^*.

3.4 Stage III: Prediction From Limited Inputs

The first two stages establish a generator that uses MLLM features and responds to factor-aware interventions. Stage III uses this fixed generator to optimize the MLLM pathway itself. The goal is to produce generation-compatible representations from limited observations, while complete-video observations provide a richer training mode for the same video target.

Shared Targets Across Observation Modes. We freeze the DiT at ϕ^* and jointly fine-tune the MLLM and connector, initialized from θ_0 and η_0. For each video x and description p, we use the text-only input $u^{\mathrm{t}} = p$ and the text-plus-complete-video input $u^{\mathrm{tv}} = (p, x)$. Both observation modes use MLLM-latent-only conditioning and are supervised with the same target video x.

We define the mixed-observation objective as

$$\mathcal{L}_{\mathrm{obs}}(\theta, \eta) = \mathbb{E}_{(x,p)\sim\mathcal{D}_{\mathrm{obs}}}\left[\sum_{m\in\{\mathrm{t},\mathrm{tv}\}} \pi_m \, \mathcal{L}_{\mathrm{FM}}\left(x \mid Z_{\theta,\eta}(u^m); \phi^*\right)\right], \tag{6}$$

where π_m denotes the sampling weight of observation mode m, with $\pi_{\mathrm{t}} + \pi_{\mathrm{tv}} = 1$. We minimize Eq. 6 over θ and η, while keeping ϕ^* fixed.

Prediction Through a Fixed Generator. Although the DiT is frozen, gradients still pass through it to the connector and MLLM. Under text-only observation, the MLLM pathway must produce representations that let the fixed generator reconstruct the target video's semantic, motion, and temporal content from limited evidence. Under complete-video observation, the MLLM has direct visual evidence but must express it through the same connector and generation interface. Training both modes toward a shared video target therefore adapts the MLLM pathway without changing the DiT.

3.5 Inference and Cinematic Control

At inference, FAPL supports three cinematic applications within one architecture. **(i) Generation from Scripts and References.** Given a script and optional character or scene references, the MLLM produces latents that condition the DiT through the trained connector and original interface, followed by flow-based sampling and VAE decoding. No complete target video is needed. The DiT can use MLLM latents alone or together with the original text and VAE reference conditions as separate inference configurations. **(ii) Factor-Aware Control.** We modify only the target factor's field in the structured prompt and re-encode the edited prompt through the MLLM and connector. We then replace the corresponding span in the original projected latents with the edited field's span, keeping all other spans unchanged, and condition the DiT on the resulting sequence. We evaluate both the requested change and non-target preservation. **(iii) Multi-Shot Generation.** Shots are generated independently from their descriptions and available references. Shared character and scene references promote consistency, while action and camera instructions vary by shot.

4 Experiments

4.1 Implementation

We build FAPL on UniVideo (Wei et al., 2026) with a 13B DiT and compare with the original UniVideo and Omni-Video 2 (Yang et al., 2026a) models (1.3B, 14B DiTs). Stages I and III sample text-only and text-plus-complete-video observations at a 1:1 ratio, while Stage II uses text-only inputs with factor-span replacement. We train Stages I–III for 10K, 15K, and 10K steps, respectively, using AdamW with a learning rate of 1×10^{-5}, zero weight decay, and a maximum gradient norm of 1.0. Training uses the bfloat16 (BF16) format, gradient checkpointing, a per-GPU batch size of 1, and 16 NVIDIA H200 GPUs. Inference uses 50 sampling steps and a classifier-free guidance (CFG) scale of 5.0. All three stages share 100K quality-filtered source–target video pairs (25K per factor), constructed through factor-specific selection and generation using CoinVE-200K (Long et al., 2026) as the primary source and OmniVBench's released training data (Li et al., 2026) for supplementary videos and references. Appendix C details the data preparation procedure.

4.2 Evaluation Data, Tasks, and Metrics

Test Data. CineFactor-10K comprises 10K single-shot test cases constructed from held-out source groups in CoinVE-200K (Long et al., 2026) and the training dataset released with OmniVBench (Li et al., 2026), with no source-video overlap with our training data.

Tasks. (i) Sparse-to-Video Generation. We evaluate on 5K cases using text-only prompts. (ii) Factor-Aware Editing. We evaluate on 5K cases, with 1.25K cases per factor. Each case modifies only the target factor's field in the structured prompt while retaining the other three fields unchanged. All methods receive the same modified prompt and permitted visual inputs through their standard inference interfaces. (iii) Multi-Shot Generation. We evaluate separately on the 100 sequences in the official ShotVerse-Bench (Yang et al., 2026b) test split. Each sequence comprises two to four shots in their original order, with shared character and scene references and shot-specific action and camera instructions. Appendix E details the evaluation protocols.

Metrics. (i) Temporal generation from limited inputs. Within shots, we measure action sequencing with Motion Order (Zheng et al., 2025) and camera-trajectory fidelity with Translation Error and Rotation Error (radians) (Yang et al., 2026b), using frozen π^3 (Wang et al., 2025c) for trajectory recovery. To complement these action/camera measures, multi-shot evaluation assesses character and scene consistency as shot-specific instructions vary, using Subject Consistency, Scene Consistency (Wang et al., 2025b), and FaceSim-Arc for facial identity (Yuan et al., 2025). (ii) Factor-aware control. We evaluate single-factor edits using FiVE's $\cap$-Acc (Li et al., 2025), requiring both yes/no and multiple-choice checks to pass. Human evaluation (see Appendix F) additionally assesses target changes and non-target preservation.

Table 1: **Quantitative comparison across three evaluation tasks.**

Method	Sparse-to-Video			Factor-Aware Editing	Multi-Shot Consistency		
	Motion Order (%)↑	Trans. Error↓	Rot. Error (rad)↓	∩-Acc (%)↑	Subject↑	Scene↑	FaceSim-Arc↑
Omni-Video 2 (1.3B)	33	0.1084	1.87	28	0.435	0.373	0.412
Omni-Video 2 (14B)	43	0.0817	1.49	37	0.476	0.423	0.468
UniVideo (13B)	48	0.0763	1.54	42	0.462	0.427	0.451
FAPL (Ours)	**61**	**0.0586**	**1.02**	**58**	**0.516**	**0.471**	**0.523**

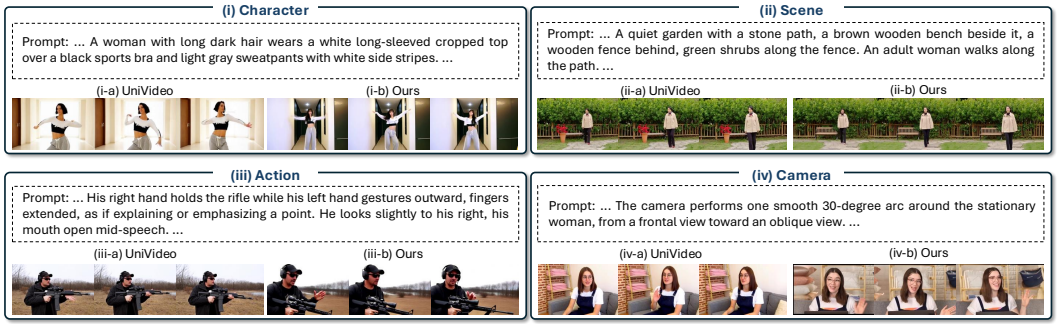

Figure 4: **Under limited inputs, baseline models fail to predict temporal visual details.** With only text and no visual references, we compare UniVideo with FAPL (Ours), highlighting representative baseline limitations across four cinematic factors. **(i) Character.** Long hair is generated as short hair, failing appearance details specified only by text. **(ii) Scene.** The background is not preserved and incorrect content appears. **(iii) Action.** The requested sequence, *e.g.*, gesturing outward with the left hand and extended fingers, is missing. **(iv) Camera.** UniVideo ignores the instruction and keeps the camera static, while Ours orbits around the character. *FAPL better realizes prompt-specified appearance, scene, and motion from text alone.*

4.3 Comparison and Analysis

Tab. 1 compares FAPL with Omni-Video 2 and UniVideo on sparse-to-video generation, factor-aware editing, and multi-shot generation. FAPL improves action ordering and camera-trajectory fidelity from text alone, achieves higher ∩-Acc for requested factor changes, and obtains stronger Subject and Scene Consistency across shots, indicating that shot-specific instructions remain compatible with shared character and scene content. For multi-shot generation, FAPL achieves a FaceSim-Arc score of 0.523, compared with 0.412 for Omni-Video 2 (1.3B), 0.468 for Omni-Video 2 (14B), and 0.451 for UniVideo (13B). This improvement complements the Subject and Scene Consistency results, indicating stronger facial identity consistency as action and camera instructions vary across shots. In the examples shown in Figs. 4, 5, and 6, UniVideo captures the broad content but misses specified details such as hair length, hand gestures, or the requested orbit, and some edits also alter non-target factors. FAPL follows these specifications more closely while preserving unrelated factors, and maintains the shared setting and character appearance across shots. These comparisons suggest that FAPL improves instruction following and cross-shot consistency on the evaluated tasks.

4.4 Ablation Study

Tab. 2 evaluates the three training stages on UniVideo (13B). Stage I improves action ordering and reduces both camera errors relative to the pretrained model. Fig. 7 further illustrates improved generation under MLLM-only conditioning after Stage I on Omni-Video 2 (1.3B and 14B) and UniVideo (13B). For Stage II, training with factor swapping achieves 57% ∩-Acc, compared with 48% for training without factor swapping under the same paired data and optimization budget. Motion Order and camera errors do not improve over this control, suggesting that the benefit is concentrated in factor-aware editing. For Stage III, mixed-observation training improves Motion Order from 57% to 61% and reduces translation and rotation errors from 0.0629 to 0.0586 and from 1.26 to 1.02 radians, respectively, compared with text-only training. Editing accuracy remains comparable to the Stage II checkpoint, changing from 57% to 58%. These results suggest that complete-video observations during training help action ordering and camera control under prompt-only inference while retaining factor-aware editing performance.

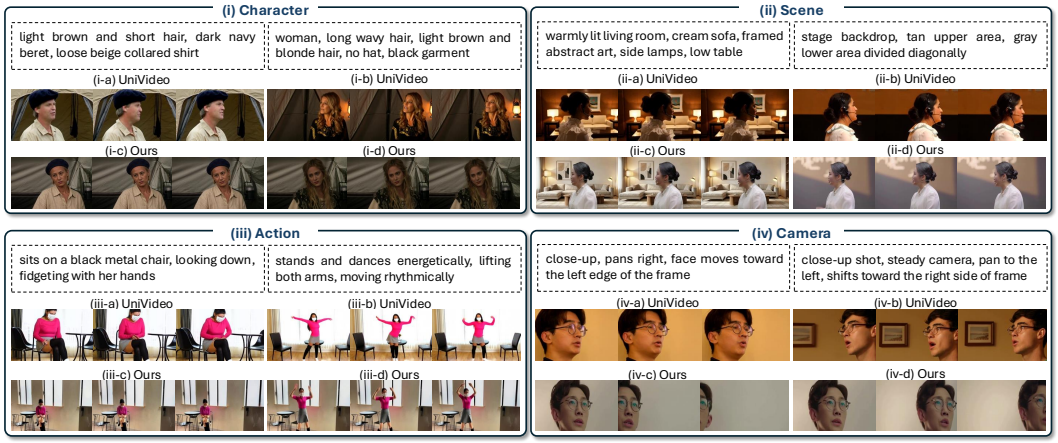

Figure 5: **FAPL improves factor-aware control.** For each factor, we modify only the corresponding description in the structured prompt and generate twice, comparing UniVideo with FAPL (Ours). UniVideo does not confine the change to the requested factor and alters unrelated content, whereas Ours modifies the selected factor while preserving the others. *This indicates that factor-level latent augmentation enables selective control.*

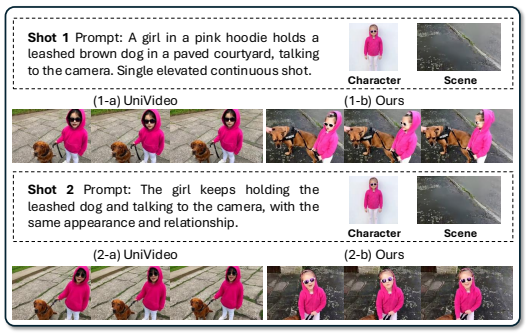

Figure 6: **FAPL supports consistent multi-shot generation without cross-shot memory.**

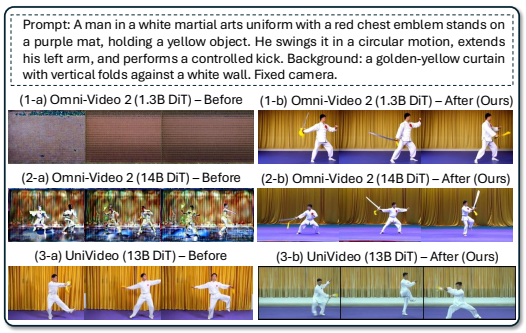

Figure 7: **MLLM latent enhancement generalizes across MLLM–DiT models.**

Table 2: **Ablation of FAPL training stages on UniVideo (13B).**

Exp Index	Training Configuration	Motion Order (%)↑	Trans. Error↓	Rot. Error (rad)↓	∩-Acc (%)↑
0	Pretrained UniVideo	48	0.0763	1.54	42
1	Stage I only	54	0.0708	1.38	44
2	Stage II: training without factor swapping (based on Exp. 1)	54	0.0654	1.30	48
3	Stage II: training with factor swapping (based on Exp. 1)	53	0.0671	1.32	57
4	Stage III: text-only training (based on Exp. 3)	57	0.0629	1.26	56
5	Stage III: (Full FAPL) mixed-observation training (based on Exp. 3)	**61**	**0.0586**	**1.02**	**58**

5 Conclusion

This work studies how MLLM representations can support temporal inference and factor-aware control in cinematic video generation. FAPL makes these representations actionable by grounding the DiT in MLLM latents, supervising factor-specific changes and preservation, and learning prediction from limited inputs. FAPL demonstrates that targeting the right bottleneck delivers substantial gains without added architectural complexity. Our findings further offer insights into how multimodal reasoning and generation can reinforce each other: reasoning guides controllable synthesis, while generation provides grounded supervision for learning to infer missing content from limited observations.

Limitations and Future Work. Although FAPL addresses a fundamental weakness of existing MLLM-conditioned DiTs by making MLLM latents more actionable, its performance remains bounded by the underlying models. Current MLLMs struggle to interpret 3D blockouts or reason about complex camera motion for DiT control, while the pretrained DiT limits visual fidelity and complex factor realization, such as realistic acting. Future work may incorporate stronger MLLMs and more capable video generators.

AI USE STATEMENT

General-purpose language-model assistants were not used to originate the method, write code, or author the paper. MLLMs used as disclosed components of data curation and automatic evaluation are described in the corresponding sections. All technical content and reported results were produced and verified by the authors, who take full responsibility for the paper.

ETHICS STATEMENT

We have read and adhere to the ICLR Code of Ethics. FAPL may inherit biases and harmful content from its pretrained models and training data, and controllable video generation may be misused to create misleading media. We encourage responsible use with appropriate consent, provenance disclosure, and safeguards against misuse.

REPRODUCIBILITY STATEMENT

The method and three-stage training procedure are detailed in Sec. 3 and Algorithm 1. Sec. 4 specifies the backbone, data, training schedule, hyperparameters, hardware, inference settings, benchmark, and evaluation metrics. **The appendix provides implementation details, explorations of alternative technical approaches, failure cases, and a user study. Additional video results are available in the supplementary HTML.**

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14834–14844. IEEE, 2025a.

Shuai Bai et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Gang Cheng, Xin Gao, Li Hu, Siqi Hu, Mingyang Huang, Chaonan Ji, Ju Li, Dechao Meng, Jinwei Qi, Penchong Qiao, Zhen Shen, Yafei Song, Ke Sun, Linrui Tian, Feng Wang, Guangyuan Wang, Qi Wang, Zhongjian Wang, Jiayu Xiao, Sheng Xu, Bang Zhang, Peng Zhang, Xindi Zhang, Zhe Zhang, Jingren Zhou, and Lian Zhuo. Wan-animate: Unified character animation and replacement with holistic replication, 2025. URL https://arxiv.org/abs/2509.14055.

Mohamed Elmoghany, Ryan Rossi, Seunghyun Yoon, Subhojyoti Mukherjee, Eslam Mohamed Bakr, Puneet Mathur, Gang Wu, Viet Dac Lai, Nedim Lipka, Ruiyi Zhang, et al. A survey on long-video storytelling generation: architectures, consistency, and cinematic quality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7082–7094, 2025.

Haopeng Fang, Di Qiu, Binjie Mao, and He Tang. Motioncharacter: Fine-grained motion controllable human video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3831–3839, 2026.

Sharath Girish, Tsai-Shien Chen, Zhikang Dong, Mukesh Singhal, Hao Chen, Sergey Tulyakov, and Aliaksandr Siarohin. Cineorchestra: Unified entity-centric conditioning for cinematic video generation. *arXiv preprint arXiv:2606.13768*, 2026.

Kaiyi Huang, Yukun Huang, Yu Li, Jianhong Bai, Xintao Wang, Zinan Lin, Xuefei Ning, Jiwen Yu, Yu Wang, and Xihui Liu. Cinescene: Implicit 3d as effective scene representation for cinematic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25381–25392, 2026a.

Yuyang Huang, Yabo Chen, Wenrui Dai, Ziyang Zheng, Haibin Huang, Chi Zhang, Junni Zou, Hongkai Xiong, and Xuelong Li. Cineweaver: Training-free reference-controllable multi-shot long video generation for cinematic storytelling. *arXiv preprint arXiv:2607.26529*, 2026b.

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 (ICCV)*, pp. 17191–17202, October 2025.

Xuan Ju, Weicai Ye, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Qiang Xu. Fulldit: Video generative foundation models with multimodal control via full attention. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15737–15747, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Minghan Li, Chenxi Xie, Yichen Wu, Lei Zhang, and Mengyu Wang. Five: A fine-grained video editing benchmark for evaluating emerging diffusion and rectified flow models. *arXiv preprint arXiv:2503.13684*, 2025.

Wenxue Li, Peiyan Guan, Haoyang Jiang, Junxian Cai, Hualuo Liu, Chunjie Zhang, Chong Guan, Kai Huang, Songlian Li, Taiyi Wu, Yongjian Yu, Xiaotong Zhao, Alan Zhao, Eric Liu, Xi Chen, Yu Liu, and Lei Zhu. Omnivbench: A benchmark and large-scale dataset for omni reference-to-video generation, 2026. URL https://arxiv.org/abs/2609.22069.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14951–14961, 2025.

Fuchen Long, Cong Wang, Zitao Gao, Wenhao Zhong, Yu Cheng, Xiaolu Hou, Yan Li, Xiao Cao, Xinlong Sun, Xi Chen, et al. Coinve-200k: A large-scale high-quality dataset for compositional instruction-guided video editing. *arXiv preprint arXiv:2608.17566*, 2026.

Yihao Meng, Hao Ouyang, Yue Yu, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Hanlin Wang, Shuailei Ma, Yixuan Li, Cheng Chen, et al. Holocine: Holistic generation of cinematic multi-shot long video narratives. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 461–471, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Liming Tan, Ye Chen, Hao Zhang, Lirong Qian, Feifei Li, and Bingbing Ni. Unimoca: Unifying motion and camera controls as visual proxies for faithful human video generation, 2026a. URL https://arxiv.org/abs/2608.01944.

Zhiyu Tan, Hao Yang, Luozheng Qin, Jia Gong, Mengping Yang, and Hao Li. Omni-video: Democratizing unified video understanding and generation, 2026b. URL https://arxiv.org/abs/2507.06119.

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.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025a.

Qinghe Wang, Xiaoyu Shi, Baolu Li, Weikang Bian, Quande Liu, Huchuan Lu, Xintao Wang, Pengfei Wan, Kun Gai, and Xu Jia. Multishotmaster: A controllable multi-shot video generation framework. *arXiv preprint arXiv:2512.03041*, 2025b.

Sophie L Wang, Phillip Isola, and Brian Cheung. The truth lies somewhere in the middle (of the generated tokens). *arXiv preprint arXiv:2605.09969*, 2026.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025c.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. 2023.

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 *The Fourteenth International Conference on Learning Representations*, 2026.

Xiaoxue Wu, Xinyuan Chen, Yaohui Wang, and Yu Qiao. Shotdirector: Directorially controllable multi-shot video generation with cinematographic transitions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2079–2089, 2026.

Junfei Xiao, Ceyuan Yang, Lvmin Zhang, Shengqu Cai, Yang Zhao, Yuwei Guo, Gordon Wetzstein, Maneesh Agrawala, Alan Yuille, and Lu Jiang. Captain cinema: Towards short movie generation. In *International Conference on Learning Representations*, volume 2026, pp. 83349–83375, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 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, 2026a. URL https://arxiv.org/abs/2602.08820.

Songlin Yang, Zhe Wang, Xuyi Yang, Songchun Zhang, Xianghao Kong, Taiyi Wu, Xiaotong Zhao, Ran Zhang, Alan Zhao, and Anyi Rao. Shotverse: Advancing cinematic camera control for text-driven multi-shot video creation. *arXiv preprint arXiv:2603.11421*, 2026b.

Songlin Yang, Haobin Zhong, Ruilin Zhang, Xiaotong Zhao, Shuai Li, Kai Zheng, Xuyi Yang, Zhe Wang, Zhenchen Tang, Yang Li, et al. Evalverse: Pipeline-aware and expert-calibrated benchmarking for professional cinematic video generation. *arXiv preprint arXiv:2605.23271*, 2026c.

Zixuan Ye, Xuanhua He, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Qifeng Chen, and Wenhan Luo. Unic: Unified in-context video editing. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyang Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-preserving text-to-video generation by frequency decomposition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12978–12988, 2025.

Peixuan Zhang, Zijian Jia, Kaiqi Liu, Shuchen Weng, Si Li, and Boxin Shi. Stage: Storyboard-anchored generation for cinematic multi-shot narrative. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 659–669, 2026a.

Peixuan Zhang, Chang Zhou, Ziyuan Zhang, Hualuo Liu, Chunjie Zhang, Jingqi Liu, Xiaohui Zhou, Xi Chen, Shuchen Weng, Si Li, et al. A benchmark and multi-agent system for instruction-driven cinematic video compilation. In *European Conference on Computer Vision*, pp. 213–230. Springer, 2026b.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

APPENDIX OVERVIEW

This appendix provides methodological details, experimental protocols, and complementary analyses for Factor-Aware Predictive Learning (FAPL). Appendix A defines the four cinematic factors and details structured descriptions, variable-length projected-span replacement, and reuse of pre-trained representations. Appendix B discusses alternative designs and motivates staged adaptation, factor-span interventions, fixed versus varied factor order, and generation-space supervision through a fixed DiT. Appendix C documents the shared training dataset, including data provenance, source-level separation, factor-specific pair construction, quality filtering, structured annotation, and pre-processing for editing and mixed-observation supervision. Appendix D summarizes optimization settings, module-freezing schedules, conditioning configurations, and matched task inputs for baseline comparisons. Appendix E specifies the CineFactor-10K and ShotVerse-Bench evaluation protocols, covering action order, camera trajectories, editing accuracy, and cross-shot subject, scene, and facial identity consistency. Appendix F presents the blinded pairwise study, task-specific criteria, aggregation procedure, and human preference results. Finally, Appendix G examines limitations arising from MLLM understanding and DiT generation capacity through 3D blockouts, complex camera motion, fight scenes, and realistic character performances. It also discusses complex text–image-to-video generation with interleaved multimodal inputs and multi-character interactions as an open research direction.

The supplementary HTML provides full prompts and additional full-length video results for inspecting temporal behavior, selective factor control, and cross-shot consistency.

Appendix Contents

Appendix A: Factor Definitions and Instruction Representation 14
Appendix B: Alternative Designs and Design Rationale 15
Appendix C: Factor-Aware Data Construction 16
Appendix D: Implementation Details 19
Appendix E: Evaluation Protocol 20
Appendix F: Human Evaluation 21
Appendix G: Limitations and Failure Cases 22

A FACTOR DEFINITIONS AND INSTRUCTION REPRESENTATION

Operational Factor Definitions. We organize cinematic content into four factors to specify what an edit changes and preserves. Character covers the main subject's identity and appearance, including clothing and hair. Scene covers the environment and background layout. Action describes the subject's motion over time, and Camera describes viewpoint motion. These factors locate interventions without assuming fully disentangled representations. Tab. 3 summarizes the boundaries used for data construction and evaluation.

Preservation is defined semantically. Camera motion can change a subject's projected position, scale, and visibility while preserving its identity, action, and scene. We therefore inspect both videos in each pair rather than infer preservation from unchanged caption fields.

Structured Descriptions and Projected-Span Replacement. Each structured description follows the fixed order Character–Scene–Action–Camera. For example a, let L_a be the projected sequence length and $[s_k^a, e_k^a)$ the token interval for factor k. The span replacement in Sec. 3.3 is

$$\mathcal{R}_k(Z^a, Z^b) = \left[Z^a_{[:s_k^a]}; Z^b_{[s_k^b:e_k^b]}; Z^a_{[e_k^a:]}\right]. \tag{7}$$

The resulting sequence has length $L_a - (e_k^a - s_k^a) + (e_k^b - s_k^b)$. We replace the complete span, update its conditioning mask, and recompute batch padding. Replacement occurs after the original connector, with no additional projection. Both directions provide supervision: $\mathcal{R}_k(Z^a, Z^b)$ targets x^b, and $\mathcal{R}_k(Z^b, Z^a)$ targets x^a.

Table 3: **Factor definitions for paired supervision.** Included and excluded edits specify the single-factor training subset.

Factor	Included changes	Excluded or ambiguous changes
Character	Identity, hair, clothing, and intrinsic appearance of the main subject.	Independent-object edits, body motion, background changes, and camera motion.
Scene	Background, location, environmental layout, and background-local appearance.	Global relighting, weather, or style changes that also alter the subject.
Action	Single-person body motion, gestures, and dynamic facial performance.	Camera motion and interactions requiring major changes to other factors.
Camera	Time-varying camera extrinsics, including pan, tilt, translation, and arc motion.	Optical zoom, focal-length changes, and static reframing alone.

Reusing the Original Representation. FAPL reuses the pretrained MLLM and connector rather than introducing a separate network to map their features into explicitly disentangled factor representations. Such a network would require learning a new representation mapping while maintaining compatibility with the pretrained DiT, adding another adaptation problem to training. Constraining this representation to four factors may also discard visual information not explicitly captured by their descriptions, including fine-grained details and relationships across factors. We therefore use structured text spans to locate interventions within the existing representation and supervise their effects through paired videos. This design encourages selective changes and preservation without requiring feature orthogonality or fully disentangled representations. Contextualized spans may retain information about other fields, consistent with the coupled-factor limitations discussed in Appendix. G.

B Alternative Designs and Design Rationale

FAPL aims to make MLLM representations useful for both prediction from limited observations and factor-aware generation. Several alternative designs could support this goal, including end-to-end optimization, explicit factorized representations, conventional paired-video fine-tuning, and direct representation alignment. We discuss the trade-offs motivating our choices below. These considerations explain the design of FAPL rather than establish that every alternative is empirically inferior.

Staged Adaptation versus Joint Optimization. An end-to-end alternative would jointly update the MLLM, connector, and DiT. Such optimization allows flexible co-adaptation, but also changes both the conditioning representations and the generator that interprets them: upstream updates alter the DiT's inputs, while DiT updates alter their decoding and the gradients returned upstream. FAPL instead separates learning to use representations from learning to produce them. Stages I and II establish a generator that uses fixed MLLM features and responds to factor-level interventions; Stage III then optimizes the MLLM and connector through that fixed generator. Each phase therefore adapts to a fixed counterpart, and Stage III cannot reduce the loss through simultaneous decoder adaptation. This choice constrains co-adaptation rather than guaranteeing better optimization. Gradients still propagate through the frozen DiT to the trainable upstream modules.

Existing Representations versus an Explicitly Factorized Space. Another option is to introduce a dedicated representation network or impose feature-separation objectives for character, scene, action, and camera. Such designs could provide more explicit factor organization, but require additional architectural or representational assumptions. In particular, separation in feature space does not by itself establish selective control in generated videos, and cinematic factors are naturally coupled. FAPL instead reuses the pretrained MLLM and connector, using structured fields to locate intervention spans without assuming that their contextualized representations are fully disentangled. Factor awareness is learned through the generator's response to these interventions. The emphasis is therefore on selective changes in generated content, rather than independence between feature subspaces.

Latent Interventions versus Conventional Paired-Video Fine-Tuning. Paired editing videos can be used directly for supervised fine-tuning, with each complete description conditioning its

corresponding target. This provides useful generation supervision, but does not explicitly constrain how a localized change in the conditioning sequence should affect the output. FAPL makes this relationship explicit by replacing only the designated factor span while retaining the remaining conditioning tokens. Paired targets supervise both the intended change and preservation of non-target content. The distinction is thus how the editing relationship is used, not merely whether paired videos are available. Under matched paired data and optimization budgets, the Stage II ablation shows improved editing accuracy over fine-tuning without factor-span replacement.

Fixed versus Varied Factor Order. We use a fixed Character–Scene–Action–Camera order for structured descriptions, but factor-span replacement requires identifiable fields rather than a particular ordering. Field labels and boundaries identify the intervention spans, whose token positions vary with description length. In our experiments, changing the order or diversifying it during training yielded no clear gains in training efficiency or generation performance over the fixed order. This is consistent with the pretrained MLLM's ability to interpret explicitly labeled factors across different layouts, leaving limited benefit from order augmentation. It does not imply order-invariant representations: reordering can still change contextualized hidden states. We therefore retain a fixed order for simple annotation and span extraction, rather than as a prerequisite for factor-aware control.

Generation-Space Supervision versus Direct Representation Alignment. A further alternative is to train limited-input representations to match those obtained from complete-video observations. Such alignment can provide a direct transfer signal, but requires choosing which features should agree despite the two inputs containing different amounts of information. Moreover, feature agreement alone does not ensure that the shared information is usable by the generator. FAPL instead connects the two observation modes through shared video targets and a fixed DiT. Their representations need not be identical; both are optimized for their ability to support generation through the same interface. The Stage III ablation supports the benefit of including complete-video observations over text-only training, but does not establish superiority over explicit alignment methods, which remain a complementary direction.

Complementary Roles of the Three Stages. The three stages progressively establish effective use of MLLM features, factor-aware control, and prediction from limited observations. Stage I trains the DiT to use features from the fixed MLLM and connector, while Stage II further trains it to respond to factor-span replacements. Stage III then fixes the trained DiT and updates the MLLM and connector to produce generation-compatible representations. This separation also limits the difficulty of adapting multiple modules simultaneously. Under joint optimization, updates to the MLLM and connector change the conditioning features received by the DiT, while updates to the DiT change how those features are decoded and the gradients returned to the upstream modules. Each component must consequently adapt to a changing counterpart. By fixing the MLLM and connector in Stages I and II and fixing the DiT in Stage III, FAPL provides a stable counterpart for each phase of adaptation. In Stage III, gradients still propagate through the frozen DiT to the MLLM and connector. Algorithm 1 and Eqs. 3–6 define the corresponding objectives.

C Factor-Aware Data Construction

Data Provenance and Sample Accounting. We use CoinVE-200K (Long et al., 2026) as the primary source of videos and existing editing pairs, supplemented by videos and reference materials from the training dataset released with OmniVBench (Li et al., 2026). We select supervision by cinematic factor: multiple changes to a subject's clothing and hair can form a Character-only edit, whereas a simultaneous character and background change does not qualify. Reference–target samples must satisfy the same single-factor change and non-target preservation requirements before being used as editing pairs.

We construct the shared training dataset through factor-specific selection and generation. Character and Scene pairs are selected from existing CoinVE edits, whereas Action and Camera pairs are constructed through motion transfer and camera-controlled generation, respectively. Candidate pairs are screened for the intended target-factor change, preservation of the other three factors, and temporal and visual quality. After filtering, we retain 100K source–target pairs, with 25K pairs per factor. All three training stages use this dataset: Stages I and III use its videos and associated structured

Algorithm 2 Construction of factor-aware editing and observation data

Require: CoinVE videos and editing-pair metadata; supplementary videos and reference materials from the training dataset released with OmniVBench; held-out source IDs; four factor definitions.

Separate source groups; exclude held-out anchors and donors from training.
Decode candidates and record source IDs, frame timestamps, sizes, and durations.
for each factor k **do**
  Select eligible anchors and factor-compatible instructions.
  if k is Character or Scene **then**
    Retrieve existing CoinVE targets; reject edits spanning multiple factors.
  else if k is Action **then**
    Select a compatible donor; retarget its motion and synthesize with Wan2.2-Animate.
  else
    Synthesize the target using a retained ReCamMaster camera preset.
  end if
  Verify the target change, all three non-target factors, and video quality.
  Describe x^a in four fields; update only field k to obtain p^b.
  Apply synchronized pair preprocessing and verify that the edit remains visible.
  Retain accepted pairs and balance the final factor subsets.
end for
Form $\mathcal{D}_{\text{edit}}$ from 100K accepted pairs, with 25K pairs per factor.
Construct $\mathcal{D}_{\text{obs}}$ from the same videos and descriptions, pairing text-only and text-plus-complete-video inputs with the same target video.
return Editing pairs, observation examples, and provenance/acceptance records.

descriptions to form text-only and text-plus-complete-video observations, while Stage II uses the pairing information for factor-swapping supervision. Training and evaluation sources are separated before pair construction.

The training set contains 100K source–target pairs, with each pair counted once regardless of training direction. Both $\mathcal{D}_{\text{edit}}$ and $\mathcal{D}_{\text{obs}}$ are constructed from this shared dataset. The former uses source–target pairs for factor-swapping supervision, while the latter associates text-only input p and text-plus-complete-video input (p, x) with the same target video x.

Construction Order and Source-Level Separation. We first organize videos from both data sources into source groups, keeping known duplicate encodings, overlapping excerpts, and associated reference materials in the same partition. The source groups used to construct CineFactor-10K are held out before training-pair construction. Four factor-specific branches then select or generate candidate source–target pairs. Generated targets inherit their source partition, and action donors and other reference materials are drawn from the same partition as their anchors. Candidates are retained only when the intended factor change is visible, the other three factors are preserved, and temporal and visual quality checks pass. Accepted pairs receive structured descriptions and undergo synchronized preprocessing, followed by a check that the edit remains visible. The retained pairs form the shared training dataset: Stage II uses their pairing information for factor-swapping supervision, while Stages I and III use the same videos and descriptions for observation supervision. Algorithm 2 summarizes this procedure.

Character: Selecting Appearance Changes. We retain existing CoinVE pairs that change the main subject's identity, hair, clothing, or appearance while preserving action, scene, and camera motion. Pairs with background changes, independent-object replacement, or altered actions are rejected. Selected existing pairs reuse their original target videos.

Scene: Selecting Background-Local Changes. We retain existing CoinVE pairs that change the environment or background layout while preserving the subject and its action. Background-local appearance changes are allowed, but global weather, lighting, or style changes that also alter the subject are excluded. Selection uses the video content as well as the editing instruction.

Action: Transferring Donor Motion. We construct action targets with Wan2.2-Animate-14B. The source video provides character appearance and background, while a separate donor video provides body and facial motion. Source and donor videos are drawn from the training partitions of the

two data sources. We align donor motion to the source interval and retarget it to the subject's scale and position. The replacement-mode inputs combine the source character reference, background video, and editing mask with the donor pose and facial-motion streams.

The editing mask covers the original subject and the region occupied by the transferred motion. We select compatible single-person motions and reject outputs with residual limbs, identity changes, incomplete background recovery, or altered camera motion.

Camera: Re-Rendering with Released Trajectories. ReCamMaster (Bai et al., 2025a) renders a source video from either training data source under a prescribed camera trajectory. We use eight released example trajectories: Pan Right (ID 1), Pan Left (ID 2), Tilt Up (ID 3), Tilt Down (ID 4), Translate Up with rotation (ID 7), Translate Down with rotation (ID 8), Arc Left with rotation (ID 9), and Arc Right with rotation (ID 10). The camera subset is approximately balanced across these types. Optical zoom presets are excluded.

We retain targets that preserve character identity, action timing, and scene content despite viewpoint-dependent visibility changes. We update only the Camera caption field and reject pairs with subject deformation or inconsistent action timing. The prescribed trajectory controls synthesis; acceptance is based on the generated video. Frozen π^3 is used for evaluation, not data filtering.

Shared Acceptance Rule and Structured Annotation. Qwen3-VL-32B-Instruct (Bai et al., 2025b) evaluates paired videos with deterministic decoding. For candidate i with target factor k_i, let c_i indicate that the intended change is visible, $p_{i,g}$ indicate preservation of non-target factor g, and $q_i^{\text{temp}}, q_i^{\text{vis}}$ indicate temporal and visual acceptability. A candidate is retained only if

$$a_i = c_i \left(\prod_{g \in \mathcal{F} \setminus \{k_i\}} p_{i,g} \right) q_i^{\text{temp}} q_i^{\text{vis}} = 1. \tag{8}$$

We reject uncertain judgments. Each accepted source receives a four-field JSON description. The target description updates the edited field and retains the other three fields verbatim. If a retained field no longer describes the target video correctly, we reject the pair.

Data-Construction Design. We reuse CoinVE videos and editing targets, together with the pre-trained synthesis capabilities of Wan2.2-Animate (Cheng et al., 2025) and ReCamMaster (Bai et al., 2025a). Our construction adds factor-specific selection, source–donor recombination for Action, a camera-trajectory subset, and joint change-and-preservation filtering. Structured descriptions then locate the latent spans used by FAPL.

Annotation Template. The annotation template specifies the factor fields and acceptance checks. The JSON below illustrates the output schema for a passing Camera pair.

```
Inspect the source and target videos and the requested edit.
Describe only visible evidence. Use four factors in this order:
Character, Scene, Action, Camera.
Change only the requested factor field in the target description.
Copy the other three fields exactly, but reject the pair if those
copied descriptions would be false. Camera motion is viewpoint motion,
not the subject's motion. Do not infer a motion from one frame alone.
Return JSON with source_description, target_description, target_factor,
target_changed, preservation, temporal_consistency, visual_quality,
and rejection_reason. Uncertain checks must not be marked true.
```

```
{
  "source_description": {"Character": "...", "Scene": "...",
                         "Action": "...", "Camera": "..."},
  "target_description": {"Character": "...", "Scene": "...",
                         "Action": "...", "Camera": "..."},
  "target_factor": "Camera",
  "target_changed": true,
  "preservation": {"Character": true, "Scene": true, "Action": true},
  "temporal_consistency": true,
  "visual_quality": true,
```

Table 4: **FAPL training configuration.** Stages I and III sample text-only and text-plus-complete-video observations at a 1:1 ratio.

Setting	Stage I	Stage II	Stage III
Training steps	10K	15K	10K
Updated modules	DiT	DiT	MLLM + connector
Frozen modules	MLLM + connector	MLLM + connector	DiT
Data	$\mathcal{D}_{\text{obs}}$	$\mathcal{D}_{\text{edit}}$	$\mathcal{D}_{\text{obs}}$
Observations	Text / text+video	Text only	Text / text+video
Content condition	MLLM latents only	Swapped MLLM spans	MLLM latents only
Optimizer	AdamW; learning rate 10^{-5}; weight decay 0		
Gradient clipping	Maximum norm 1.0		
Compute	16 NVIDIA H200 GPUs; per-GPU batch size 1		
Memory / precision	Gradient checkpointing; BF16		
Inference	50 sampling steps; CFG scale 5.0		

```
  "rejection_reason": ""
}
```

This JSON illustrates the schema rather than a measured sample. Acceptance is computed from the individual change, preservation, and quality decisions.

Variable-Length, Backbone-Compatible Preprocessing. We retain the original resolution and duration of CoinVE videos and the generation metadata of synthetic targets. Training-time decoding, resizing, and cropping follow the selected backbone. For UniVideo, we use the 480p preprocessing configuration with a $(480, 854)$ resolution setting and at most 129 frames (Wei et al., 2026); final tensor dimensions follow the processor's divisibility requirements.

Clips remain variable-length. For an aligned interval containing T_{avail} frames, we choose $T = 4\lfloor(\min(T_{\text{avail}}, 129) - 1)/4\rfloor + 1$ to satisfy the temporal VAE constraint. Longer intervals are cropped contiguously, with the same temporal window and spatial transform applied to both videos. We retain crops in which the target edit remains visible and do not extend shorter clips by repeating frames. Complete-video observation gives the MLLM access to the target interval through its visual processor; the sampled MLLM frames and the DiT reconstruction frames need not be identical.

For cross-backbone Stage I experiments, we process the same data with each backbone's video and MLLM processors. Original timestamps and frame rates are retained.

D Implementation Details

Training Scope and Optimization. We train the complete FAPL pipeline and perform quantitative stage ablations on UniVideo with a 13B DiT. Tab. 4 summarizes the optimization settings. The before/after comparisons in Fig. 7 additionally evaluate Stage I on Omni-Video 2 with 1.3B and 14B DiTs.

All three stages use MLLM latents as the only content conditions. The VAE still encodes training targets and decodes generated latents, and the DiT receives the noisy video latent z_t and diffusion timestep. The pretrained flow-matching schedule is retained across stages.

Baselines and Matched Task Inputs. UniVideo (Wei et al., 2026) provides the direct backbone baseline, while Omni-Video 2 (Yang et al., 2026a) provides comparisons at 1.3B and 14B DiT scales. Each method uses its supported interface for the same task inputs. Sparse-to-video evaluation uses prompts without target frames. For factor-aware editing, all methods use the same edit specification and permitted visual inputs. Baselines process the complete edited prompt through their standard inference interfaces. FAPL instead encodes the original and edited prompts through its MLLM and connector, replaces only the target-factor span in the original projected latents with its edited counterpart, and conditions the DiT on the resulting sequence. Multi-shot generation uses shared character and scene references with shot-specific action and camera instructions. FAPL generates shots independently, without propagating generated-video context or latent state.

Table 5: **Evaluation tasks and sample counts.**

Task	Input protocol	Evaluation size
Sparse-to-video generation	Text prompt only	5K cases
Factor-aware editing	Modify one prompt field; retain the other three	5K cases; 1.25K per factor
Multi-shot generation	Shared character and scene references; per-shot action and camera instructions	100 sequences; 2–4 shots each

As described in Sec. 3.5, MLLM-latent-only inference and inference with native text or VAE-reference conditions are separate configurations. The distinction concerns the available conditioning pathways, while the task specifies which visual references are permitted.

E Evaluation Protocol

Task Organization. We evaluate on 5K sparse-to-video cases and 5K factor-aware editing cases from CineFactor-10K, which comprises 10K single-shot test cases constructed from held-out source groups in CoinVE-200K (Long et al., 2026) and the training dataset released with OmniVBench (Li et al., 2026). Source-video groups are separated from training before benchmark construction. The editing subset contains 1.25K cases for each of Character, Scene, Action, and Camera. Multi-shot evaluation uses 100 sequences from the ShotVerse-Bench (Yang et al., 2026b) test split. Each sequence comprises two to four shots in their original order, with shared character and scene references and shot-specific action and camera instructions. These sequences are used only for evaluation. All methods use the same sequences and permitted inputs. Table 5 summarizes the input protocols and evaluation sizes.

Motion Order. We adapt Motion Order Understanding from VBench-2.0 (Zheng et al., 2025). Qwen3-VL-32B-Instruct (Bai et al., 2025b) uses deterministic decoding to check whether the specified actor performs the required actions in order. For binary decisions o_i over N samples, Motion Order is $100N^{-1}\sum_i o_i$. The check evaluates the ordered events rather than action-category recognition alone.

Camera Errors. Following ShotVerse (Yang et al., 2026b), we recover trajectories from generated and target videos using frozen π^3 (Wang et al., 2025c). Trajectories are expressed relative to the first frame and evaluated with Translation Error and Rotation Error. Rotation Error is measured in radians.

Factor-aware Editing. We follow the FiVE protocol (Li et al., 2025) used in the main paper. An edit passes only when both the yes/no and multiple-choice checks succeed. Denoting their binary outcomes by y_i^{YN} and y_i^{MC}, $\cap$-Acc is defined as

$$\cap\text{-Acc} = \frac{100}{N}\sum_{i=1}^{N} y_i^{\mathrm{YN}} y_i^{\mathrm{MC}}. \tag{9}$$

Question Content. Target questions check the requested change, while preservation questions compare each non-target factor in the After video with the same method's generated Before video. For a Camera edit, the three checks concern character identity and appearance, scene identity and layout, and action content and order. Viewpoint-induced visibility changes alone do not count as scene changes. We retain the multiple-choice questions and distractors from the original evaluation.

Inter-Shot Consistency. Following MultiShotMaster (Wang et al., 2025b), Subject Consistency and Scene Consistency measure inter-shot feature similarity rather than agreement with a reference image. We use its subject and scene extraction and aggregation protocol, with each generated sequence as the reporting unit.

Facial Identity Consistency. We use FaceSim-Arc, as in ConsisID (Yuan et al., 2025), to evaluate whether the same character preserves their facial identity across generated shots. Subject Consistency measures broader subject-level similarity, whereas FaceSim-Arc specifically evaluates facial identity under changes in action, viewpoint, and camera motion. Our cross-shot evaluation compares faces between generated shots rather than comparing each generated shot with an external reference image.

For sequence n, the generated sequence contains $K_n \in \{2, 3, 4\}$ shots. Let $\mathcal{A}_{n,i}$ denote the set of valid face observations of the evaluated character in shot i of sequence n. For each detected and aligned face crop, we extract an ArcFace embedding and apply ℓ_2 normalization, obtaining a unit-length feature vector $\mathbf{e}_{n,i,a}$ for observation $a \in \mathcal{A}_{n,i}$. We compute the similarity between two different shots by averaging the cosine similarities over all cross-shot pairs of valid face observations:

$$s^{\text{face}}_{n,ij} = \frac{\sum_{a \in \mathcal{A}_{n,i}} \sum_{b \in \mathcal{A}_{n,j}} \mathbf{e}^{\top}_{n,i,a} \mathbf{e}_{n,j,b}}{|\mathcal{A}_{n,i}|\,|\mathcal{A}_{n,j}|}, \qquad i < j, \quad |\mathcal{A}_{n,i}| > 0, \quad |\mathcal{A}_{n,j}| > 0. \tag{10}$$

Only observations of the same evaluated character are compared. Because the embeddings have unit length, their inner product equals cosine similarity. Within-shot comparisons are excluded.

For sequence n, let $\mathcal{V}_n = \{i : |\mathcal{A}_{n,i}| > 0\}$ denote the shots containing valid face observations of the evaluated character, and let $M_n = |\mathcal{V}_n|$. We compute similarities only between valid shots. Sequences with fewer than two valid shots are excluded from FaceSim-Arc evaluation. For each remaining sequence, we average over all unordered pairs of valid shots:

$$S^{\text{face}}_n = \frac{1}{\binom{M_n}{2}} \sum_{\substack{i,j \in \mathcal{V}_n \\ i<j}} s^{\text{face}}_{n,ij}. \tag{11}$$

Let $\mathcal{E}_{\text{face}} = \{n : M_n \geq 2\}$ denote the eligible sequences. The final score averages their sequence-level scores:

$$\text{FaceSim-Arc} = \frac{1}{|\mathcal{E}_{\text{face}}|} \sum_{n \in \mathcal{E}_{\text{face}}} S^{\text{face}}_n. \tag{12}$$

This procedure assigns equal weight to each valid shot pair within a sequence and to each eligible sequence. Exclusions apply only to this metric. We report raw cosine similarity without percentage scaling. The theoretical range is $[-1, 1]$, with higher scores indicating greater facial identity consistency across shots.

F Human Evaluation

Study Protocol. We compare FAPL with its UniVideo (13B) baseline using a blinded pairwise study. Following the task-oriented assessment in ShotVerse (Yang et al., 2026b), we evaluate instruction following, consistency, and visual quality for the three tasks. Tab. 6 lists the criteria.

The study includes 20 participants and 24 comparison cases: eight sparse-to-video cases, eight factor-editing cases with two per factor, and eight multi-shot sequences. Each participant evaluates all cases using anonymous A/B labels, randomized method placement, and randomized case order. Participants receive the task prompt and permitted references and can replay the videos. Editing trials show each method's generated Before and After videos. Multi-shot trials present all shots in their original order.

Participants select one of three responses: "A better," "tie," or "B better." They judge each stated criterion rather than treating greater motion or larger visual changes as inherently better.

Aggregation and Reporting. We aggregate responses separately for each task and question. Each question receives 160 ratings per task from 20 participants evaluating eight cases. After mapping the randomized A/B labels back to the methods, we count preferences for FAPL, ties, and preferences for UniVideo. Each percentage is computed by dividing the corresponding response count by 160 and multiplying by 100. The four questions are reported separately rather than combined into a composite score. Tab. 7 reports the three task-specific criteria and overall preference for each task. Repeated ratings from the same participants and cases are not independent observations.

Table 6: **Human-evaluation criteria.** Participants answer each question by selecting A better, tie, or B better.

Task	Questions: which result better satisfies the criterion?
Sparse-to-video	(1) Requested content and attributes; (2) action order and camera behavior; (3) natural, temporally stable motion; (4) overall preference given the instruction.
Factor-aware editing	(1) Successful target-factor change; (2) preservation of all three non-target factors; (3) natural, temporally stable motion; (4) overall preference given the edit.
Multi-shot	(1) Character consistency across shots; (2) scene consistency across shots; (3) each shot's action/camera instructions; (4) overall sequence preference.

Table 7: **Human preference by evaluation criterion (%).** Participants compare FAPL with UniVideo (13B) using three choices: A better, tie, or B better, with method identities hidden. Each row summarizes 160 responses. Percentages may not sum to 100 due to rounding.

Evaluation criterion	FAPL preferred	Tie	UniVideo preferred
Sparse-to-video generation			
Content and attribute alignment	55.6	18.1	26.3
Action order and camera alignment	59.4	13.1	27.5
Motion naturalness and temporal stability	48.8	21.9	29.4
Overall preference	56.9	16.9	26.3
Factor-aware editing			
Target-factor change	63.1	12.5	24.4
Non-target factor preservation	58.8	18.1	23.1
Motion naturalness and temporal stability	51.9	19.4	28.8
Overall preference	61.3	15.6	23.1
Multi-shot generation			
Character consistency across shots	56.3	18.1	25.6
Scene consistency across shots	53.1	23.1	23.8
Per-shot action and camera alignment	57.5	15.6	26.9
Overall preference	54.4	19.4	26.3

Tab. 7 reports preferences for individual criteria and overall results. FAPL receives more preferences than UniVideo across all evaluated criteria, with the highest rate for target-factor change in editing (63.1%). Preferences for non-target preservation and cross-shot character and scene consistency also favor FAPL. The margins are smaller for motion naturalness and temporal stability, suggesting that the preference advantage is more pronounced for instruction following and consistency than for this quality criterion. Overall preference rates are 56.9%, 61.3%, and 54.4% for sparse-to-video generation, factor-aware editing, and multi-shot generation, respectively. These judgments complement the automatic metrics on the evaluated cases.

G Limitations and Failure Cases

FAPL improves how MLLM representations condition video generation, but its capabilities remain bounded by the information captured by the MLLM and the visual behavior that the DiT can synthesize. We discuss four challenging settings under two broad categories: multimodal understanding and video generation.

MLLM Understanding Limitations. **(a) 3D Blockouts.** Blockout references convey scene geometry and spatial arrangement while omitting much of the texture, material, and appearance information available in natural videos. Interpreting them requires identifying simplified objects and inferring their spatial and temporal relationships from limited visual cues. Ambiguities in these references can lead to incorrect interpretations of object roles, depth ordering, or motion, which may subsequently affect the generated scene. Training the generator to use MLLM features more effectively cannot recover reference information that those features fail to capture. **(b) Complex**

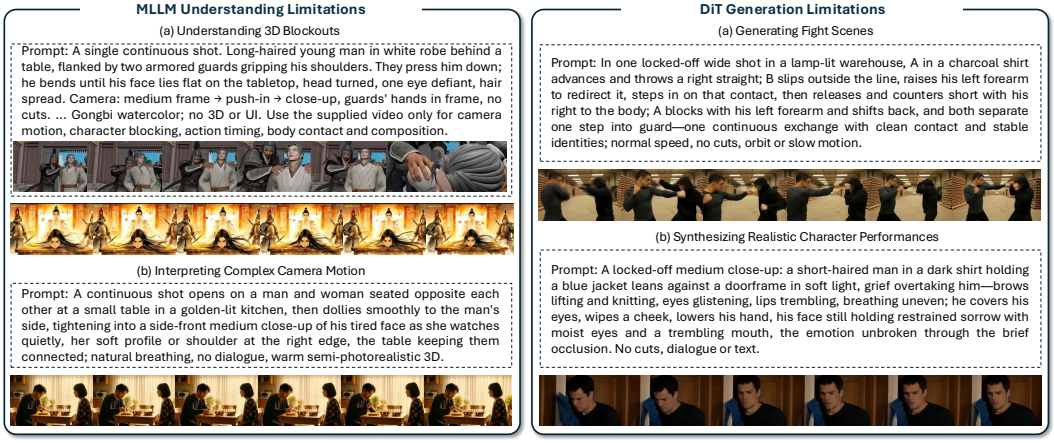

Figure 8: **Failure Cases: Limits of MLLM Understanding and DiT Generation.**

Camera Motion. Interpreting complex camera instructions in a prompt requires strong spatial understanding to reason about the camera's position, orientation, and motion relative to the subject and scene. Instructions combining translation and rotation, curved paths, or sequential changes in direction require more than recognizing individual motion terms: the MLLM must infer how these movements compose into a coherent trajectory over time. Its representations may capture the broad intent while failing to encode these spatial and temporal relationships accurately. Consequently, the generated video may simplify the requested trajectory or execute only part of the instruction. These cases motivate stronger spatial reasoning in the MLLM to support precise camera control.

DiT Generation Limitations. **(a) Fight Scenes.** Fight scenes require coordinated motion between multiple characters, rapid changes in pose, and consistent handling of contact and occlusion. Even when the intended interaction is clearly specified, synthesizing it requires maintaining plausible body configurations and synchronizing the participants throughout the sequence. Failures can take the form of missed contact, inconsistent reactions, or distorted limbs during fast motion. Such interactions also couple several factors: an action necessarily changes relative positions and visibility, so preserving non-target content must remain compatible with these physical consequences. **(b) Realistic Character Performances.** Convincing performance depends on subtle coordination among facial expressions, gaze, gestures, posture, and timing. A generated character may perform the requested action at a semantic level while lacking the fine-grained transitions that make it appear natural. Factor-aware supervision can encourage adherence to the requested behavior, but does not by itself ensure expressive detail or realistic motion. These settings therefore expose limits in the fidelity and temporal coordination of the underlying video generator, beyond whether the requested factor is represented in its conditions.

Complex Text–Image-to-Video Generation. Extending FAPL to complex text–image-to-video generation remains an open direction, particularly when inputs interleave textual instructions and visual references involving multiple characters and intricate interactions. Such inputs require the MLLM to associate references with the correct entities, resolve cross-modal correspondences, and infer spatial and temporal relationships. The DiT must then realize these relationships while preserving distinct identities and coordinating motion, contact, and occlusion. FAPL improves how MLLM representations guide generation, but does not remove these underlying understanding and synthesis bottlenecks. Our current setting does not establish reliable handling of such complex interleaved inputs, and single-factor paired supervision does not fully capture coordinated multi-character interactions. Addressing these scenarios will require further research into multimodal grounding, interaction-aware supervision, and the capabilities of both the MLLM and video generator.