IMAGINE IN DISCRETE SPACE: DISCRETE VISUAL LATENT REASONING FOR VIDEO EVENT PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video event prediction requires multimodal large language models (MLLMs) to infer events beyond an observed video segment. We investigate whether future-video supervision can support this task through discrete visual intermediate states. We propose **Imagine in Discrete Space (IDS)**, which quantizes future-video features into visual tokens and learns to predict these tokens autoregressively from observed videos. A three-stage pipeline learns future-token prediction, grounds quantized future tokens in event descriptions, and refines event prediction with reinforcement learning. At inference, only the observed segment is available as visual input, and the final prediction is conditioned on self-generated visual tokens. The formulation provides a next-token training interface for future visual features without a pixel-reconstruction objective. We also introduce **VEPBench**, a benchmark of approximately 3K samples with recorded and plausible alternative futures and typed distractors, alongside a source-disjoint 17K-video training set. On VEPBench, IDS before reinforcement learning achieves 67.00% recorded-future selection, compared with 54.41% for the text-reasoning variant using the same backbone and reported SFT budget. The full model reaches 72.78%. Results on FutureBench provide evidence of cross-benchmark transfer, with stronger performance on one-hop prediction than on multi-hop tasks. These findings support discrete future visual tokens as a useful intermediate representation in the evaluated setting.

1 INTRODUCTION

Multimodal large language models (MLLMs) can recognize actions, locate temporal evidence, and answer questions about observed videos. Video event prediction (VEP) extends this setting to events that have not yet been observed: given a video prefix, a model predicts what is likely to happen next. This requires relating the observed actions, participants, and scene context to possible continuations. A recorded continuation supplies a concrete evaluation target, but it is not necessarily the only plausible future.

Recent approaches improve multimodal reasoning through textual chain-of-thought (CoT) (Shao et al., 2024a), visual latent states (Yang et al., 2025; Wang et al., 2025c; Li et al., 2025), and reinforcement learning with verifiable rewards (Feng et al., 2025; Wang et al., 2025b). Textual reasoning can express temporal and commonsense relations, but an explicit textual intermediate representation may omit visual details relevant to a subsequent event. Visual intermediate states offer a complementary way to retain feature-level supervision from future videos.

Continuous latent reasoning provides one way to introduce such states. Some implementations supervise intermediate features with distance-based losses in addition to answer-level cross-entropy. IDS explores a categorical formulation of future-feature supervision that shares the next-token training interface of the MLLM. We ask: *Can future-video features be learned as discrete, autoregressively predicted intermediate states that improve event prediction in an MLLM?* We evaluate this question through comparisons of intermediate representations and training configurations.

We propose **Imagine in Discrete Space (IDS)**. A visual codebook quantizes features extracted from future video segments into a sequence of discrete tokens. An MLLM first learns to predict these to-

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.

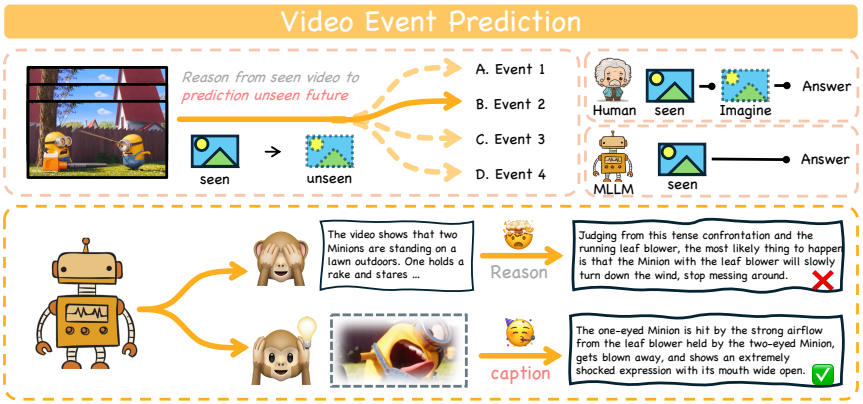


Figure 1: Video event prediction from an observed segment. IDS predicts discrete future visual features as intermediate states before producing an event prediction. Future video is used for supervision during training only.

kens from the observed video, then learns to map quantized future tokens to event descriptions, and finally refines event prediction through reinforcement learning. At inference, the model generates its own visual tokens from the observed segment before producing a textual answer. Here, "imagination" denotes prediction of future-oriented features, not synthesis of video pixels or a verified simulation of future dynamics.

Discrete video representation and autoregressive prediction are established ideas (Yan et al., 2021). IDS uses them to connect future-video supervision to event prediction within an MLLM. Its contribution is the use of these tokens as supervised intermediate states for a downstream semantic prediction, rather than a new quantization algorithm. Because future-token prediction uses cross-entropy, it can share the model's next-token training interface. Whether this design is beneficial remains an empirical question.

We also introduce **VEPBench**, a benchmark of approximately 3K samples that separates selection of the recorded future from selection of either of two designated plausible continuations. Typed distractors provide additional information about the kinds of alternatives selected by a model. A separate 17K-video training set uses different source datasets. This separation addresses source overlap, while semantic overlap and shared option-generation patterns require separate consideration.

Our experiments prioritize comparisons among Qwen2.5-VL-7B variants. Before reinforcement learning, IDS improves recorded-future selection from 54.41% for textual reasoning to 67.00% under the reported common SFT budget; the full pipeline reaches 72.78%. FutureBench results extend the evaluation beyond VEPBench, with gains that are more pronounced for one-hop prediction. Training curves and feature visualizations provide additional descriptive analyses. We will release all data and code to facilitate future research.

Our contributions are:

- A future-supervised visual intermediate representation for VEP, implemented as discrete tokens that an MLLM predicts before generating an event answer.
- A three-stage training pipeline combining future-token prediction, event-description grounding, and task-level reinforcement learning, evaluated against text and continuous-latent variants.
- VEPBench and a separate training set, with protocols that distinguish recorded-future agreement, selection of designated plausible futures, and typed distractor selection.

2 Related Work

Video Event Prediction. Video event prediction aims to infer plausible future events from observed video context, where the target is unobserved and thus requires temporal understanding,

causal abstraction, and commonsense reasoning Liang et al. (2025); Lei et al. (2020). Prior work studies future-event prediction Lai et al. (2025); Su et al. (2025); Wang et al. (2025a); TEMPURA uses masked-event prediction for temporal reasoning Cheng et al. (2025). Related textual script-prediction methods model event dependencies through compositional representations, event graphs, and generative training Granroth-Wilding & Clark (2016); Li et al. (2018); Zhu et al. (2023); VidEvent extends structured event modeling to videos Liang et al. (2025). With the emergence of MLLMs, recent approaches adapt pretrained vision-language models to VEP through supervised fine-tuning or reinforcement learning Su et al. (2026); Wang et al. (2025a). MVP Tan et al. (2023) learns forecasting representations by predicting future features across timescales. VLEP Lei et al. (2020), V1-33K, and FutureBench Wang et al. (2025a) provide complementary datasets and evaluation settings. VEPBench adds a structured candidate protocol that distinguishes recorded-future agreement from designated plausible alternatives and reports typed distractor selection. Such a design also requires caution about option-construction artifacts, as demonstrated in TemporalBench (Cai et al., 2024).

Vision-Language Model Reasoning. Multimodal reasoning extends textual CoT through visual operations, interleaved evidence, and reinforcement learning. Representative approaches use visual search and editing (Wu & Xie, 2024; Fu et al., 2025; Liu et al., 2024), diffusion-based latent features (He et al., 2024), or knowledge- and task-guided CoT (Mondal et al., 2024; Luo et al., 2024). Video reasoning methods also use reinforcement learning to improve task performance (Feng et al., 2025; Wang et al., 2025d;b). IDS investigates a complementary source of supervision: future visual features used as intermediate targets for event prediction.

Latent Reasoning in MLLMs. Latent reasoning studies whether models can perform intermediate reasoning in hidden representations instead of relying entirely on explicit textual rationales Biran et al. (2024); Yang et al. (2024). Early work compresses reasoning traces into implicit hidden states through rationale distillation Deng et al. (2023), or introduces special thought tokens to encourage computation before generation Wang et al. (2023); Goyal et al. (2023). Recent methods further replace discrete intermediate text with continuous latent embeddings, suggesting that reasoning can be internalized beyond natural-language chains Hao et al. (2024). Extending latent reasoning to multimodal settings, however, remains challenging. Chameleon and Anole support interleaved image–text generation Chameleon Team (2024); Chern et al. (2024); this capability alone does not demonstrate reasoning through generated images. Others adopt continuous visual embeddings for latent visual reasoning Yang et al. (2025); Zhang et al. (2025); Wu et al. (2026). IDS studies discrete future-feature tokens as an alternative intermediate representation within an MLLM; the comparison does not imply a general incompatibility between continuous states and autoregressive models.

Discrete Video Representations. VideoGPT (Yan et al., 2021) combines vector-quantized video representations with autoregressive modeling for video generation. Perception Tokens (Bigverdi et al., 2024) uses discrete visual tokens for intermediate perceptual representations. IDS instead quantizes future-video encoder features for event prediction, without a pixel decoder. Its distinction is future-supervised intermediate prediction within an MLLM, not discrete visual reasoning alone.

3 Benchmark and Training Data

Evaluation rationale. VEP evaluation must distinguish agreement with a recorded continuation from plausibility under an observed context. These are related but different targets: a plausible prediction need not match the single future captured in a video. VEPBench uses approximately 3K samples and a five-option protocol to report both targets within a specified candidate set, together with typed distractor selection. It is a diagnostic evaluation of future-event prediction, rather than an exhaustive characterization of the distribution of possible futures.

Candidate structure. Given an observed segment $\mathcal{V}$, a model selects one of five candidates: the recorded future (A), a plausible alternative (B), an observed historical event (C), a subject-substituted event (D), and an action-inconsistent future (E). The construction probes three distinct departures from the target continuation: copying history, replacing the subject, and changing the action inconsistently with context. These labels describe candidate construction; selection of a candidate alone does not uniquely identify the model's internal failure mechanism. In particular, B is an accepted alternative for the plausibility metric, not an ordinary incorrect distractor.

Construction and verification. VEPBench is constructed through a model-assisted pipeline followed by manual verification of all evaluation samples. For each sample, we split a VEP-suitable video clip into an observed segment and a subsequent future segment. The future segment supports reference annotation and candidate generation and is withheld from evaluated prediction models. It is also available to the external judges in open-set evaluation. As detailed in Appendix A, candidates are generated according to the five definitions, and the evaluation set is selected by model-assigned difficulty strata before human verification. This organizes coverage within the collected pool; it does not establish population representativeness or independently calibrated difficulty.

All selected samples are manually verified for recorded-future correctness, alternative-future plausibility, and distractor validity; samples with incorrect recorded futures, implausible alternatives, or invalid distractors are corrected or discarded. More than 90% of the reviewed samples require no correction, and more than 95% of Option-B candidates are judged as plausible future continuations. During evaluation, models receive only the observed segment, and candidate order is randomized before metric computation.

Metrics and interpretation. We report the **selection rate** (SR) of the canonical candidate types. $\mathrm{SR@A}$ measures agreement with the recorded future. $\mathrm{SR@AB}$ accepts either A or B and measures selection of the two designated plausible continuations; it does not measure predictive calibration or coverage of all plausible futures. $\mathrm{SR@C/D/E}$ report the corresponding distractor-selection rates. A lower rate for one distractor is not sufficient evidence of improvement, since selections may shift to another distractor. We therefore interpret these rates jointly with the primary metrics.

We additionally remove the candidates and evaluate free-form predictions against the recorded future with external visual judges. These open-set scores are judge-assigned semantic-agreement scores, not multiple-choice accuracies, and their magnitudes are not directly comparable to SR. Candidate randomization addresses position bias; open-set evaluation removes candidate access at inference. Neither alone excludes stylistic or semantic regularities learned from the data-construction pipeline (Cai et al., 2024).

Separate training data. We construct a separate 17K-video training set from sources not used in VEPBench. Its construction centers on open-ended future prediction, using observed video context and subsequent events to provide visual and semantic supervision. Stages 1 and 2 learn future visual tokens and event descriptions without answer candidates. Stage 3 subsequently uses multiple-choice tasks for reinforcement learning. Thus, the open-ended construction describes the supervised learning stages, not a candidate-free training pipeline; source separation does not imply semantic or domain disjointness. Figure 3 summarizes the benchmark distribution; Appendix A provides further details.

4 METHODS

Figure 2 contrasts the intermediate representations studied in this work: text, continuous visual states, and discrete visual tokens. IDS uses quantized future features as token-level supervision, allowing future-feature prediction to be trained with cross-entropy. This design choice does not presuppose that the alternative representations are unsuitable for VEP.

We propose **Imagine in Discrete Space (IDS)** to perform future-event reasoning through discrete visual tokens. IDS initializes a visual codebook from the MLLM vision encoder and quantizes future-video representations into token-compatible visual states. These tokens serve as future-oriented intermediate states, allowing the model to reason in a visual-semantic space with an NTP objective in the supervised stages. We instantiate IDS with a three-stage training pipeline, as shown in Figure 4, that progressively builds future visual imagination, semantic alignment, and final event prediction. The discrete visual space and each training stage are detailed in the following sections.

4.1 INITIALIZING THE DISCRETE VISUAL SPACE

IDS requires a token-compatible visual vocabulary to represent future-oriented visual states. In our implementation, the future segment is first encoded by the frozen MLLM vision encoder and converted into a sequence of visual embeddings following the encoder's native embedding procedure.

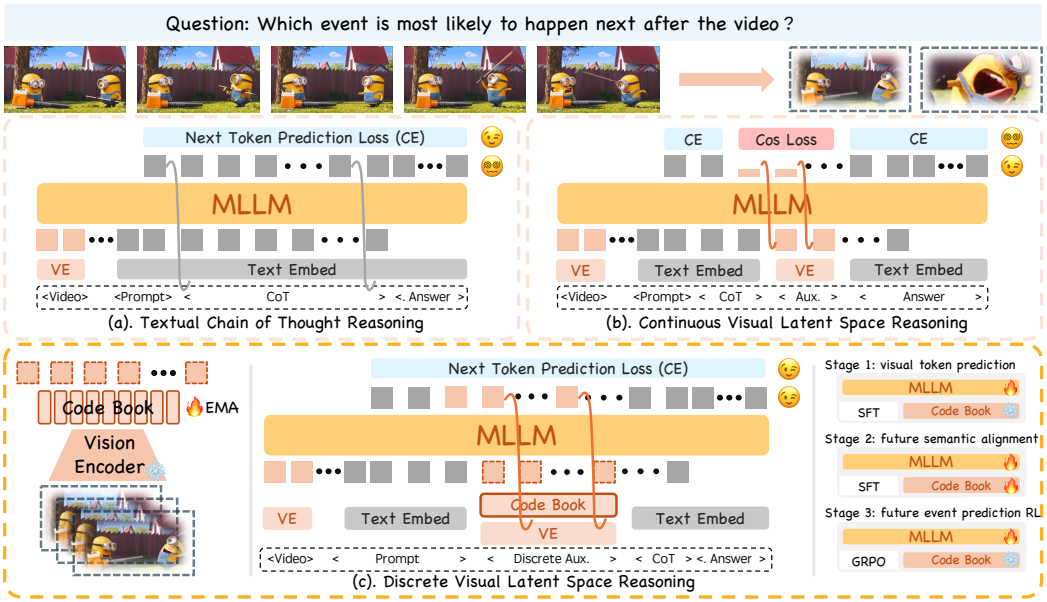


Figure 2: Comparison of reasoning paradigms for VEP. (a) Textual CoT performs future-event prediction through verbalized intermediate rationales, which may compress fine-grained visual-temporal dynamics into language. (b) The evaluated continuous latent variant combines feature-distance supervision with answer-level NTP. (c) IDS quantizes future videos into discrete visual tokens through a codebook, enabling visual latent reasoning to be optimized with the standard autoregressive next-token objective.

Each visual embedding corresponds to one selected visual token, and therefore captures a compact feature-level abstraction of the future segment rather than a pixel-level reconstruction target. We keep the sequence length as $T = 84$ by truncation or padding with the last embedding, so that the sequence has a fixed budget during autoregressive prediction.

Given the resulting continuous future-video embeddings $\mathcal{E} = \{e_n\}_{n=1}^{N}$, each embedding is assigned to its nearest codebook entry in $\mathcal{C} = \{q_i\}_{i=1}^{s}$, producing a discrete sequence $I_{1:T}$. We then introduce new visual tokens $\{\langle I_k \rangle\}_{k=1}^{s}$ into the MLLM vocabulary and initialize their embeddings with the corresponding codebook vectors. EMA is used only for codebook initialization, before MLLM training (Appendix D). Unlike VQ-VAE-style reconstruction, our codebook is not trained to reconstruct pixels; it provides a compact token-compatible abstraction of future-video features for downstream VEP.

4.2 Stage 1: Learning to Imagine Future Visual Tokens

The first stage trains the MLLM to imagine future visual states in the discrete latent space. Given an observed video and a stage-specific instruction prompt, the model predicts the corresponding future visual token sequence $I_{1:T}$ obtained from the quantized future-event segment.

Each training sample is organized as:

$$S^{(1)} = (Q_{1:M}^{(1)}, I_{1:T}), \tag{1}$$

where $Q_{1:M}^{(1)}$ denotes the observed video input together with the instruction prompt, and $I_{1:T}$ denotes the discrete future visual tokens. We optimize only the future-token positions with the NTP loss:

$$\mathcal{L}_1 = -\sum_{t=1}^{T} \log p_\theta \left(I_t \mid Q_{1:M}^{(1)}, I_{<t} \right). \tag{2}$$

Stage 1 predicts codebook indices using standard autoregressive cross-entropy and gradient back-propagation through the MLLM, without further EMA updates. Its empirical effect is evaluated in Section 5.

4.3 Stage 2: Grounding Future Tokens in Event Descriptions

Stage 2 is designed to semantically ground the discrete visual tokens learned in Stage 1. Training conditions event-description generation on the observed input and quantized ground-truth future tokens. At inference, IDS predicts visual tokens from the observed input and uses them to condition the final answer. Each training sample is formatted as:

$$S^{(2)} = (Q^{(2)}_{1:M}, I_{1:T}, D_{1:H}), \tag{3}$$

where $Q^{(2)}_{1:M}$ denotes the observed video and prompt, $I_{1:T}$ denotes the discrete future visual tokens, $D_{1:H}$ denotes the textual description of the future event. The loss is applied to the description tokens; the conditioning sequence contains quantized ground-truth future features rather than model-generated future tokens:

$$\mathcal{L}_2 = -\sum_{t=1}^{H} \log p_\theta \left(D_t \mid Q^{(2)}_{1:M}, I_{1:T}, D_{<t} \right). \tag{4}$$

The quantized future-token sequences are precomputed and held fixed throughout Stage 2; no re-quantization is performed. The visual-token embeddings are updated with the MLLM through the description-token loss, while the discrete indices remain unchanged.

The use of ground-truth future tokens in this stage introduces a conditioning difference from inference, where the tokens are generated from the observed video. Stage 3 trains on sampled model responses, but the present experiments do not separately quantify this conditioning gap.

4.4 Stage 3: Reinforcement Learning for Future-Event Prediction

The first two stages enable the model to generate future-oriented visual tokens and align them with textual event semantics. However, VEP ultimately requires the model to use these imagined visual states to make the correct future-event prediction. We therefore apply GRPO Shao et al. (2024b) to multiple-choice prediction, with rewards for answer accuracy and output validity.

Given an observed video input $Q^{(3)}$, the model first generates discrete visual tokens and then produces the final textual prediction. For each input, we sample a group of G responses from the old policy:

$$\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid Q^{(3)}), \tag{5}$$

where each response y_i contains the generated reasoning trajectory and the final answer.

Each response receives a reward composed of an accuracy reward r_i^{acc} and a format reward r_i^{fmt}:

$$r_i = \lambda r_i^{\text{acc}} + (1-\lambda) r_i^{\text{fmt}}. \tag{6}$$

The accuracy reward is defined based on the parsed final answer. In the main experiments, a response receives full reward only when it selects the recorded future event, while plausible alternative futures are not assigned additional accuracy reward. We also explore a softened reward variant that assigns a partial reward $\eta \in [0, 1)$ to the plausible alternative future event. This variant changes the relative reward assigned to the two designated plausible continuations. Its effect is evaluated through selection rates rather than coverage of multiple futures. Table 6 reports the corresponding reward ablation (Appendix H). The format reward encourages the model to follow the required output structure, including producing the imagined visual-token sequence and the final answer in the specified format.

We then update the model with the clipped GRPO objective and a KL penalty to the reference policy, with details provided in Appendix E. During Stage 3, the codebook and visual-token embeddings are frozen; the remaining trainable MLLM parameters are optimized to refine the prediction policy.

Table 1: VEPBench results. SR reports selection percentages for canonical types A–E; open-set reports mean scores (0–100) from Gemini 3.1 Pro and GPT-5.4. Best results are bold; second-best are underlined. * marks primary metrics; – indicates unavailable scores. NL: non-latent baselines trained on the same VEP data as IDS.

Model / backbone	Method	Selection rate (%)					Open-set ↑	
		A↑*	AB↑*	C↓	D↓	E↓	Gemini	GPT
Commercial Models								
Gemini 3.1 Pro	Vanilla	51.79	58.65	34.76	5.93	0.65	28.70	30.59
GPT-5.4	Vanilla	52.06	63.43	30.76	**4.00**	1.80	23.03	27.56
Open-Source Models								
Qwen2.5-VL-3B	Vanilla	36.65	57.53	24.00	14.78	3.70	8.57	10.36
Qwen2.5-VL-7B	Vanilla	41.38	65.75	27.94	4.80	1.50	33.53	46.08
Qwen2.5-VL-32B	Vanilla	54.48	72.47	20.03	5.00	2.46	30.03	47.06
Qwen3-VL-8B	Vanilla	39.93	60.67	32.07	5.93	1.33	28.12	30.58
Qwen3-VL-32B	Vanilla	53.80	63.94	31.40	<u>4.02</u>	<u>0.64</u>	31.96	37.12
MiMo-VL-7B	Vanilla	36.83	67.47	24.53	5.60	2.40	33.13	37.54
LLaVA-NeXT-Video-7B	Vanilla	52.97	70.33	15.83	8.40	5.43	23.90	34.53
InternVL3-8B	Vanilla	42.88	68.79	21.71	7.44	2.07	27.70	32.07
Kimi-VL-A3B	Vanilla	46.48	<u>74.38</u>	15.81	6.70	3.11	18.80	24.35
GLM-4.6V-Flash-9B	Vanilla	51.40	59.78	**7.26**	9.50	23.46	25.98	30.12
Keye-VL-8B	Vanilla	38.77	60.40	24.07	10.87	4.67	33.90	<u>49.28</u>
Visual Reasoning								
Video-R1	Vanilla	47.93	71.53	19.80	7.17	1.50	<u>35.00</u>	47.11
Video-Chat-A1	Vanilla	41.17	64.17	26.50	7.10	2.23	33.60	37.67
Qwen2.5-VL-7B	SFT^{NL}	54.41	69.41	12.17	11.53	6.90	31.20	45.18
Qwen2.5-VL-7B	GRPO^{NL}	51.37	64.54	14.07	13.60	7.80	33.13	45.96
Latent Reasoning								
Qwen2.5-VL-7B	LVR	55.41	68.40	<u>11.17</u>	12.53	7.90	–	–
Qwen2.5-VL-7B	Monet	53.41	70.41	<u>11.17</u>	10.53	7.89	–	–
Qwen3-VL-8B	SwimBird	<u>56.41</u>	68.41	13.17	10.52	7.90	–	–
Qwen2.5-VL-7B	Mirage	49.63	61.48	14.01	14.28	10.27	32.03	44.27
Qwen2.5-VL-7B	**IDS (ours)**	**72.78**	**76.62**	13.41	9.41	**0.57**	**37.07**	**49.48**

We evaluate predictive performance on VEPBench and transfer to FutureBench, then examine training stages, codebook size, and intermediate representations.

5 Experiments

Cross-family comparisons provide context; training curves and feature projections provide descriptive analyses.

Controlled comparisons use the same training samples and matched training budgets and comparable hyperparameters, including training duration and update counts. Comparisons with and without RL are distinguished. SFT and GRPO are non-latent controls: they train answer generation on the same data as IDS without latent visual intermediate states, enabling comparison with latent reasoning. Appendix B provides metrics and implementation settings; Table 5 reports sensitivity to the input-frame budget.

5.1 Main Results

Table 1 compares VEPBench performance; the controlled Qwen2.5-VL-7B variants provide the primary evidence for IDS.

Table 2: Evaluation on FutureBench. Higher is better. Best results are bold; second-best are underlined. NL: non-latent baselines trained on the same VEP data as IDS.

Model / backbone	Method	FutureBench ↑				
		1-Hop	2-Hop	3-Hop	Interp.	AVG
Commercial Models						
GPT-4o	Vanilla	61.90	61.70	<u>72.10</u>	51.60	59.04
GPT-5		59.60	57.30	62.60	55.60	57.92
Open-Source Models						
Qwen2.5-VL-3B	Vanilla	45.10	50.80	44.30	45.80	46.30
Qwen2.5-VL-7B		57.20	57.00	50.20	50.70	52.94
Qwen2.5-VL-32B		<u>66.50</u>	62.70	63.20	55.20	59.94
Qwen2.5-VL-72B		55.50	<u>68.40</u>	63.70	53.20	58.33
MiMo-VL-7B		59.00	59.60	50.50	43.80	50.45
LLaVA-NeXT-Video-7B		48.80	49.30	40.00	44.40	45.24
InternVL3-8B		54.30	58.00	63.20	54.40	56.72
GLM-4.1V-9B		29.90	41.90	52.20	47.30	44.38
Kimi-VL-A3B		44.30	42.80	51.30	51.90	48.87
Qwen3-VL-30B-A3B		65.30	**70.50**	**76.10**	**62.20**	**66.86**
Visual and Latent Reasoning						
Qwen2.5-VL-7B	SFT^{NL}	52.02	56.99	57.21	49.49	52.75
Qwen2.5-VL-7B	GRPO^{NL}	63.01	56.48	52.24	50.92	54.17
Qwen2.5-VL-7B	Mirage	53.18	53.37	49.75	45.19	48.86
Qwen2.5-VL-7B	LVR (SFT+RL)	22.5	26.4	22.9	17.6	$21.0^{\dagger}$
Qwen2.5-VL-7B	Monet (SFT+RL)	46.8	47.2	45.3	49.7	47.9
Qwen3-VL-8B	SwimBird (SFT)	59.0	66.8	64.7	<u>61.8</u>	<u>62.8</u>
Qwen2.5-VL-7B	**IDS (ours)**	**69.36**	60.62	56.22	58.08	60.04

IDS leads on both primary VEPBench metrics, SR@A and SR@AB. These comparisons support the complete pipeline but do not isolate discretization from training-stage and objective differences. It surpasses the strongest SR@A baseline, SwimBird-8B, by 16.37 points, and improves over the best competing SR@AB result, Kimi-VL-A3B, by 2.24 points. Although comparisons with larger open-source and proprietary models are not strictly like-for-like training comparisons, they provide useful contextual reference. IDS also obtains the highest reported open-set means under both judges. These scores evaluate agreement with the recorded future without exposing answer candidates; they do not establish absence of training-data shortcuts or statistical significance. The diagnostic rates do not improve uniformly: IDS has a higher SR@C than the SFT baseline (13.41 versus 12.17), while reducing SR@D and SR@E.

On FutureBench (Table 2), IDS achieves the best 1-Hop accuracy and the third-best average score. SwimBird exceeds IDS on 2-Hop, 3-Hop, interpolation, and average score. Cross-benchmark transfer should also be distinguished from verified domain-disjoint generalization.

LVR (Li et al., 2025) and Monet (Wang et al., 2025c) use Qwen2.5-VL-7B, as does IDS; SwimBird (Tong et al., 2026) uses Qwen3-VL-8B. This newer backbone may contribute to SwimBird's FutureBench advantage, but quantifying that contribution requires a matched-backbone evaluation. Shared backbones alone also do not control training data, supervision, or compute.

Our Mirage reproduction (Yang et al., 2025) uses the same textual data and future-video supervision as IDS. This comparison is relevant to the choice of intermediate representation, but differences in objective and implementation remain.

5.2 Ablation on Training Stage and Codebook Size

Table 3 reports the stage-wise results for codebook sizes 200 and 400. The Stage 1-only model generates visual tokens and then continues to predict the answer, without access to ground-truth future tokens. These stage-wise comparisons evaluate training configurations; results with and without RL

should be distinguished when interpreting gains. Diagnostic rates need not improve monotonically. For example, with 200 entries, SR@E rises from 3.43 to 4.41 after Stage 3.

Table 3: Ablations on codebook size and the three-stage training pipeline.

Stage	SR@A↑*	SR@AB↑*	SR@C↓	SR@D↓	SR@E↓
Codebook Size 200					
1	35.87	60.34	21.20	12.15	6.31
1 + 2	58.45	68.68	16.73	11.16	3.43
1 + 2 + 3	63.21	70.32	15.06	10.21	4.41
Codebook Size 400					
1	37.21	59.63	32.55	**5.64**	2.18
1 + 2	67.00	72.35	14.12	12.35	1.18
1 + 2 + 3	**72.78**	**76.62**	**13.41**	9.41	**0.57**

The 400-entry setting is stronger on both primary metrics after Stage 2 and Stage 3, but codebook size alone does not establish greater utilization, better semantic organization, or resistance to collapse.

5.3 Comparison of Intermediate Representations

Table 4 compares three variants with the same backbone and SFT budget. **Textual Caption** describes the future before answering; **Textual Reasoning** uses language-only CoT; and **Visual Reasoning** predicts discrete visual tokens before generating the textual answer. The Visual Reasoning variant corresponds to IDS before the final GRPO stage.

Table 4: Ablations on VEP reasoning patterns.

Method	SR@A↑*	SR@AB↑*	SR@C↓	SR@D↓	SR@E↓
Textual Caption	44.06	62.44	28.52	**7.33**	1.71
Textual Reasoning	54.41	69.41	**12.17**	11.53	6.90
Visual Reasoning (IDS)	**67.00**	**72.35**	14.12	12.35	**1.18**

Before GRPO, Visual Reasoning improves over Textual Reasoning by 12.59 points on SR@A (54.41 to 67.00) and 2.94 points on SR@AB (69.41 to 72.35), but has higher SR@C and SR@D. Thus, gains on the primary metrics do not imply uniformly lower distractor selection.

These results support the usefulness of future-supervised visual states but do not isolate discretization from supervision and computation. Figure 5 compares projected visual-token, historical-video, and future-event text features. Figure 6 shows smoother IDS training curves in the reported runs, although batch-size and logging differences limit attribution to discretization. These analyses characterize representations and optimization behavior; neither projected proximity nor loss-direction mismatch establishes causal mediation or gradient conflict (Appendix F).

6 Conclusion

IDS combines future-token prediction, event-description grounding, and reinforcement learning. VEPBench separates recorded-future agreement, designated plausible-continuation selection, and typed distractor rates. Results support the pipeline and transfer to FutureBench, especially for one-hop prediction, without establishing universal superiority of discrete states or attributing all gains to discretization.

Limitations. Source-disjoint construction does not exclude semantic overlap or shared generation artifacts. Open-set judge scores complement candidate-based evaluation but do not replace human evaluation of model predictions. The current comparisons do not fully disentangle representation, supervision, and computation; codebook utilization and inference cost also require further characterization. Finally, single-continuation supervision and the present metrics do not establish coverage of the distribution of plausible futures.

AI USE STATEMENT

Generative AI tools were used for model-assisted benchmark construction and model-based evaluation, as described in Appendix A, and to assist manuscript revision, citation checks, formatting, interpretation of results, methodological critique, and suggestions for additional experiments. Suggestions for additional experiments are not reported as completed studies. This statement describes the uses documented in the present manuscript and revision process.

REPRODUCIBILITY STATEMENT

The training objectives are provided in the method section and Appendix E. Appendix D describes codebook initialization, Appendix B lists implementation settings, and Appendix I provides the construction, training, and evaluation prompts. These descriptions document the reported procedure; further details needed to reproduce the data selection and implementation remain to be supplied.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. *arXiv preprint arXiv:2412.03548*, 2024.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. *arXiv preprint arXiv:2406.12775*, 2024.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. TemporalBench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024. URL https://arxiv.org/abs/2410.10818.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2024. URL https://arxiv.org/abs/2405.09818.

Jen-Hao Cheng, Vivian Wang, Huayu Wang, Huapeng Zhou, Yi-Hao Peng, Hou-I Liu, Hsiang-Wei Huang, Kuang-Ming Chen, Cheng-Yen Yang, Wenhao Chai, et al. Tempura: Temporal event masked prediction and understanding for reasoning in action. *arXiv preprint arXiv:2505.01583*, 2025. URL https://arxiv.org/abs/2505.01583v1.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. Refocus: Visual editing as a chain of thought for structured image understanding. *arXiv preprint arXiv:2501.05452*, 2025.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Mark Granroth-Wilding and Stephen Clark. What happens next? event prediction using a compositional neural network model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30 (1), 2016.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Liqi He, Zuchao Li, Xiantao Cai, and Ping Wang. Multi-modal latent space learning for chain-of-thought reasoning in language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18180–18187, 2024.

Kimi Team. Kimi-VL technical report, 2025. URL https://arxiv.org/abs/2504.07491.

Kwai Keye Team. Kwai keye-vl technical report, 2025. URL https://arxiv.org/abs/2507.01949.

Chenghang Lai, Weifeng Ge, and Xiangyang Xue. Cross-modal complementary learning and template-based reasoning chains for future event prediction in videos. *IEEE Transactions on Multimedia*, 2025.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. What is more likely to happen next? video-and-language future event prediction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 8769–8784, 2020.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning, 2025. URL https://arxiv.org/abs/2509.24251.

Zhongyang Li, Xiao Ding, and Ting Liu. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*, 2018.

Baoyu Liang, Qile Su, Shoutai Zhu, Yuchen Liang, and Chao Tong. Videvent: A large dataset for understanding dynamic evolution of events in videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5):5128–5136, 2025.

Zuyan Liu, Yuhao Dong, Yongming Rao, Jie Zhou, and Jiwen Lu. Chain-of-spot: Interactive reasoning improves large vision-language models. *arXiv preprint arXiv:2403.12966*, 2024.

Xuewen Luo, Fan Ding, Yinsheng Song, Xiaofeng Zhang, and Junnyong Loo. Pkrd-cot: A unified chain-of-thought prompting for multi-modal large language models in autonomous driving. *arXiv preprint arXiv:2412.02025*, 2024.

Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. *Proceedings of the AAAI conference on artificial intelligence*, 38(17):18798–18806, 2024.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning, 2024a. URL https://arxiv.org/abs/2403.16999.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b. URL https://arxiv.org/abs/2402.03300.

Qile Su, Shoutai Zhu, Shuai Zhang, Baoyu Liang, and Chao Tong. Eventformer: A node-graph hierarchical attention transformer for action-centric video event prediction. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 4698–4707, 2025. URL https://arxiv.org/abs/2510.21786.

Qile Su, Jing Tang, Rui Chen, Lei Sun, and Xiangxiang Chu. Video-coe: Reinforcing video event prediction via chain of events. *arXiv preprint arXiv:2603.14935*, 2026.

Reuben Tan, Matthias De Lange, Michael Iuzzolino, Bryan A Plummer, Kate Saenko, Karl Ridgeway, and Lorenzo Torresani. Multiscale video pretraining for long-term activity forecasting. *arXiv preprint arXiv:2307.12854*, 2023.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

Jintao Tong, Shilin Yan, Hongwei Xue, Xiaojun Tang, Kunyu Shi, Guannan Zhang, Ruixuan Li, and Yixiong Zou. SwimBird: Eliciting switchable reasoning mode in hybrid autoregressive MLLMs. *arXiv preprint arXiv:2602.06040*, 2026. URL https://arxiv.org/abs/2602.06040.

Haonan Wang, Hongfu Liu, Xiangyan Liu, Chao Du, Kenji Kawaguchi, Ye Wang, and Tianyu Pang. Fostering video reasoning via next-event prediction. *arXiv preprint arXiv:2505.22457*, 2025a.

Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning, 2025b. URL https://arxiv.org/abs/2505.12434.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language, 2025c. URL https://arxiv.org/abs/2511.21395.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. *arXiv preprint arXiv:2310.05707*, 2023.

Zikang Wang, Boyu Chen, Zhengrong Yue, Yi Wang, Yu Qiao, Limin Wang, and Yali Wang. Videochat-a1: Thinking with long videos by chain-of-shot reasoning, 2025d. URL https://arxiv.org/abs/2506.06097.

Linquan Wu, Tianxiang Jiang, Yifei Dong, Haoyu Yang, Fengji Zhang, Shichaang Meng, Ai Xuan, Linqi Song, and Jacky Keung. Lavit: Aligning latent visual thoughts for multi-modal reasoning. *arXiv preprint arXiv:2601.10129*, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

Xiaomi LLM-Core Team. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021. URL https://arxiv.org/abs/2104.10157.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

Z.ai. GLM-4.6V-Flash model card, 2025. URL https://huggingface.co/zai-org/GLM-4.6V-Flash. Accessed September 26, 2026.

Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, et al. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in MLLMs. *arXiv preprint arXiv:2510.24514*, 2025. URL https://arxiv.org/abs/2510.24514.

Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model, 2024. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/.

Fangqi Zhu, Jun Gao, Changlong Yu, Wei Wang, Chen Xu, Xin Mu, Min Yang, and Ruifeng Xu. A generative approach for script event prediction via contrastive fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):14056–14064, Jun. 2023. doi: 10.1609/aaai.v37i11.26645. URL https://ojs.aaai.org/index.php/AAAI/article/view/26645.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. URL https://arxiv.org/abs/2504.10479.

A Details of VEPBench Construction

A.1 Option Definitions

VEPBench follows a five-option multiple-choice formulation. Given an observed video segment $\mathcal{V}$, the model selects one candidate from $\{O_i\}_{i=1}^{5}$. The canonical option types are defined as follows.

Option A: Recorded future event. This option describes the actual event that occurs immediately after the observed segment. SR@A measures strict recorded-future prediction.

Option B: Plausible alternative future. This option describes an event that does not occur in the original video but remains logically plausible given the observed context. Together with Option A, it is used to compute SR@AB, which measures whether the model can identify plausible future continuations rather than only the single recorded outcome.

Option C: Historical-event distractor. This option describes an event that has already occurred in the observed segment and is distinct from the future event. SR@C reports selection of historical-event distractors.

Option D: Subject-substituted event. This option replaces the recorded future event's subject with another entity, requiring general commonsense plausibility but not compatibility with the observed context. SR@D reports selection of subject-substituted candidates.

Option E: Action-inconsistent future event. This option keeps the main participants of the recorded future event but replaces the action with one that is inconsistent with the observed event dynamics. SR@E reports selection of action-inconsistent distractors.

During evaluation, candidate order is randomized for each sample, and the canonical option type is restored only for metric computation.

A.2 Open-set Evaluation Protocol

Beyond the closed-set multiple-choice setting, VEPBench includes an open-set protocol for free-form future-event prediction. In this setting, all answer candidates are removed, and the model is asked to directly describe the event that is most likely to happen next based only on the observed video segment.

Let $\hat{y}_i$ denote the generated prediction for the i-th test sample. We use external visual judges to compare $\hat{y}_i$ with the recorded future segment and assign a score $s_i \in [0, 100]$ according to semantic consistency with the recorded future visual evidence:

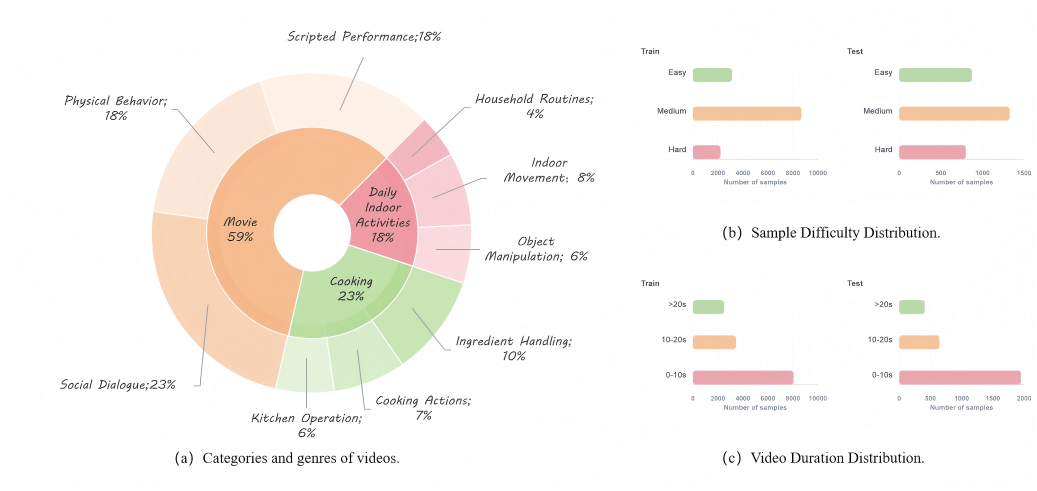
$$\text{Open-set} = \frac{1}{N} \sum_{i=1}^{N} s_i.$$

The judges are given the observed segment, the recorded future segment, and the model-generated prediction, but are not given the closed-set answer candidates or the model identity. This protocol measures recorded-future semantic agreement without candidate access. It does not exhaustively score plausible futures, rule out learned construction artifacts, or provide a quantity directly comparable to closed-set selection percentages. Judge preferences may affect the scores.

A.3 Data Construction Pipeline

We construct VEPBench through a model-assisted generation pipeline followed by human verification. For the evaluation set, we first collect candidate videos suitable for VEP and split each video into an observed segment and a subsequent future segment. The future segment is used for reference annotation and candidate generation. It is withheld from evaluated prediction models but provided to external judges in open-set evaluation.

For each candidate sample, we use a strong MLLM to generate Options A–E according to the diagnostic definitions above and to assign a difficulty score based on the dependency between the observed events and the future event. An easy sample has clear visual cues and a direct temporal or logical connection between the observed segment and the future event; the target event still occurs



(a) Categories and genres of videos.

(b) Sample Difficulty Distribution.

(c) Video Duration Distribution.

Figure 3: Data distribution statistics of VEPBench. (a) shows the category distribution of all video data. (b) shows difficulty distributions and (c) shows duration distributions; each includes the training and test sets.

after the observed segment. We then stratify the candidate pool by difficulty and select approximately 3K samples to form the final VEPBench evaluation set. All selected samples are manually verified before inclusion.

This pipeline separates future-grounded annotation, structured candidate generation, difficulty-based selection, and human verification. These components serve different purposes: annotation supplies the recorded reference; candidate types specify the diagnostic contrasts; stratification organizes the selected pool; verification checks candidate validity. Model-assigned difficulty is not an independently validated measure of human difficulty, and the selected set should not be treated as a random sample of all video events.

A.4 Separate Training Data

To train IDS, we separately construct a 17K-video VEP training set from source datasets that are not used to build VEPBench. The training and evaluation sets are kept disjoint to prevent direct train–test data leakage. It does not imply that the event categories, scenarios, or semantic content are disjoint. When video identifiers or metadata are available, we additionally remove duplicated or near-duplicated videos. The training set is constructed around open-ended future prediction. Observed segments provide context, while subsequent segments supply visual features and event descriptions for supervision. Stages 1 and 2 use candidate-free supervision to learn future-event representations and descriptions; Stage 3 uses multiple-choice reinforcement learning. The stage-specific use of the training signals is described in Section 4 and Appendix E. The difference in construction should not be interpreted as proof of semantic or domain disjointness; VEPBench additionally provides an open-set evaluation protocol.

A.5 Data Statistics

Figure 3 summarizes the data distribution of our dataset. The videos cover diverse event domains and fine-grained subcategories, including movies, cooking, and daily indoor activities. We also report the difficulty and duration distributions for both the training and test sets, showing that the dataset spans different reasoning difficulty levels and temporal lengths.

A.6 Human Verification

To ensure benchmark quality, human annotators manually review all VEPBench evaluation samples (approximately 3K). For each sample, annotators check whether: (1) Option A correctly describes

the recorded future event; (2) Option B is a logically plausible continuation under the observed context; (3) Option C is a valid historical-event distractor; (4) Option D satisfies the subject-substitution definition; and (5) Option E reflects an action-level event-evolution failure. Samples with incorrect recorded futures, implausible alternative futures, or invalid distractors are corrected or discarded before finalization. More than 90% of the reviewed samples require no correction, and more than 95% of Option B candidates are judged as plausible future continuations.

B Experimental Details

Model references. The published LVR and Monet models use Qwen2.5-VL-7B; SwimBird uses Qwen3-VL-8B. Tables identify these backbones separately from methods. SwimBird reports stronger Qwen3-VL-8B results than Qwen2.5-VL-7B on several general visual benchmarks, but this does not establish a backbone-only explanation of the FutureBench gap. The present VEPBench vanilla results also do not show a uniform Qwen3 advantage (SR@A: 39.93 versus 41.38). LVR (Li et al., 2025), Monet (Wang et al., 2025c), and SwimBird (Tong et al., 2026) identify the additional latent-reasoning baselines in Tables 1 and 2. The model families in Table 1 include Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), MiMo-VL (Xiaomi LLM-Core Team, 2025), LLaVA-NeXT-Video (Zhang et al., 2024), InternVL3 (Zhu et al., 2025), Kimi-VL (Kimi Team, 2025), GLM-V (Team et al., 2025), and Keye-VL (Kwai Keye Team, 2025). Video-R1 (Feng et al., 2025), Video-Chat-A1 (Wang et al., 2025d), and Mirage (Yang et al., 2025) provide additional reasoning comparisons. The GLM-4.6V-Flash model card (Z.ai, 2025) specifies 9B parameters. Model-family references do not replace checkpoint identifiers and evaluation configurations.

B.1 Benchmarks and Metrics

VEPBench. We evaluate models on the VEPBench test set of approximately 3,000 samples following the diagnostic protocol defined in Section 3. For closed-set evaluation, we report the selection rate of each candidate option. Higher $\mathrm{SR@A}$ indicates greater recorded-future agreement; $\mathrm{SR@AB}$ additionally accepts the designated plausible alternative. $\mathrm{SR@C/D/E}$ report typed distractor selection and should be interpreted jointly with the primary metrics, rather than as direct measures of internal failure mechanisms.

Open-set evaluation. For open-set evaluation, all answer candidates are removed and models are asked to directly generate the most likely future event from the observed video. We use GPT-5.4 and Gemini 3.1 Pro as visual judges and report the average judged score over VEPBench.

FutureBench. We further evaluate IDS on FutureBench as an external future-event reasoning benchmark to assess cross-benchmark generalization. FutureBench reports 1-Hop, 2-Hop, 3-Hop, interpolation, and an aggregate score.

- **1-Hop:** predict one future event connecting the observed scenes to the final scene.
- **2-Hop:** infer two consecutive future events connecting the observed scenes to the final event.
- **3-Hop:** predict three consecutive future events, extending the required event sequence.
- **Interpolation:** infer non-consecutive events from partially observed scenes that include intermediate anchors.

B.2 Implementation Details

We use Qwen2.5-VL-7B as the backbone for all in-house variants and train our models using up to 4 NVIDIA A100 GPUs. During training, we sample at most 32 video frames and cap the visual resolution at $128 \times 28 \times 28$ for efficiency. Unless otherwise specified, all trainable variants use the same optimization budget for fair comparison: a learning rate of 1×10^{-6} and 1,000 training steps.

For GRPO, the group size G is set to 4, the KL coefficient β is set to 0.04, and the clipping parameter ϵ is set to 0.2.

C Effect of Input Frame Number

Table 5 studies the effect of input frame numbers using the same Qwen3-VL-8B model.

Table 5: Frame-number ablation on Qwen3-VL-8B. The best results are highlighted in **bold**.

Frames	SR@A↑	SR@AB↑	SR@C↓	SR@D↓	SR@E↓
4	36.03	56.70	35.63	6.80	**0.87**
8	38.20	59.47	33.00	6.37	1.17
16	39.93	**60.67**	**32.07**	5.93	1.33
32	**41.17**	60.30	32.37	**5.87**	1.47

Increasing the number of frames generally improves SR@A, suggesting that additional visual-temporal context helps identify the recorded future event. The 16-frame setting achieves the best SR@AB and the lowest SR@C, while 32 frames further improves SR@A but brings only marginal changes on other metrics. Therefore, we report the 16-frame result for this model in Table 1 to keep the main comparison consistent with the default evaluation setting.

This frame-number experiment characterizes input sensitivity for the specified backbone. It does not test IDS scalability or isolate the effect of its intermediate representation.

D Details of Discrete Visual Codebook Initialization

This section describes EMA-based codebook initialization before MLLM training. EMA updates are not used during Stages 1–3. The goal of the codebook is to convert continuous future-video representations into token-compatible discrete visual states, rather than reconstructing the original video pixels.

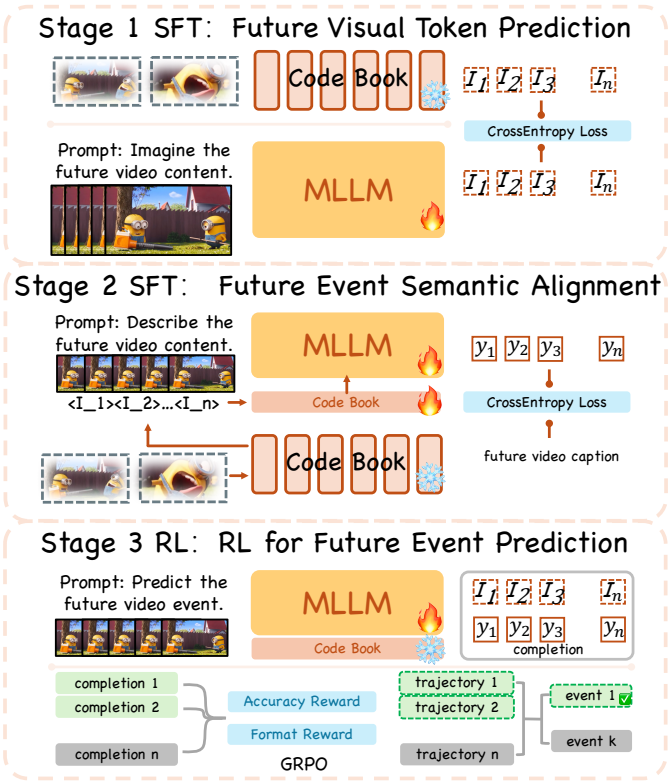


Figure 4: Three-stage training pipeline of IDS. The figure illustrates the input-output examples and supervision methods used at each stage.

Future-video embeddings. Given a future video segment used for training supervision, we extract its continuous visual embeddings with the frozen MLLM vision encoder. Let

$$\mathcal{E} = \{e_n\}_{n=1}^{N}, \quad e_n \in \mathbb{R}^d, \tag{7}$$

where N denotes the number of visual embeddings and d denotes the embedding dimension. We initialize a discrete visual codebook as

$$\mathcal{C} = \{q_i\}_{i=1}^{s}, \quad q_i \in \mathbb{R}^d, \tag{8}$$

where s is the codebook size.

Nearest-neighbor assignment. For each continuous visual embedding e_n, we assign it to the nearest codebook entry according to the squared Euclidean distance:

$$z_n = \arg\min_{i \in \{1,\dots,s\}} \|e_n - q_i\|_2^2. \tag{9}$$

This produces a sequence of discrete codebook indices. After applying the assignment to all future-video embeddings, each future segment is represented as a discrete visual-token sequence:

$$I_{1:T} = (I_1, I_2, \dots, I_T), \tag{10}$$

where each I_t corresponds to a codebook index.

EMA updates during initialization. For the i-th codebook entry, let $\mathcal{E}_i^{(t)} = \{e_n \mid z_n = i\}$ denote the set of embeddings assigned to this entry in the current minibatch at update step t, and let $n_i^{(t)} = |\mathcal{E}_i^{(t)}|$ be the corresponding cluster size. We maintain two exponential moving averages: the cluster count $N_i^{(t)}$ and the embedding sum $m_i^{(t)}$. They are updated as:

$$\begin{aligned} N_i^{(t)} &= \gamma N_i^{(t-1)} + (1-\gamma) n_i^{(t)}, \\ m_i^{(t)} &= \gamma m_i^{(t-1)} + (1-\gamma) \sum_{e_n \in \mathcal{E}_i^{(t)}} e_n, \\ q_i^{(t)} &= \frac{m_i^{(t)}}{N_i^{(t)} + \delta}, \end{aligned} \tag{11}$$

where $\gamma \in [0, 1)$ is the EMA decay factor and δ is a small constant for numerical stability. This EMA rule updates assignment counts and feature sums rather than applying a reconstruction-loss gradient to the centers. It does not by itself guarantee balanced utilization or prevent inactive codes.

Mapping codebook entries to MLLM tokens. After obtaining the codebook, we introduce a set of new visual tokens

$$\{\texttt{<I_k>}\}_{k=1}^{s} \tag{12}$$

into the MLLM vocabulary. The embedding of each token `<I_k>` is initialized with the corresponding codebook vector q_k. For Stage 2, the future-token index sequences are precomputed and fixed. The description loss updates visual-token embeddings without recomputing these index assignments. In this way, a quantized future-video representation can be expressed as a standard token sequence and optimized with the same autoregressive NTP objective used by MLLMs.

Difference from reconstruction-based quantization. IDS differs from VQ-VAE-style reconstruction methods in both objective and usage. The codebook is not trained to reconstruct pixels, and no decoder or pixel-level reconstruction loss is introduced. Instead, the codebook serves as a discrete visual latent space for reasoning: it maps continuous future-video features into discrete visual tokens that can be predicted, aligned with textual event semantics, and used as intermediate states for future-event prediction.

E Details of GRPO Training in Stage 3

This section provides the detailed GRPO objective used in Stage 3. As described in Section 4.4, Stage 3 further optimizes IDS for the final VEP task after the model has learned to generate future-oriented visual tokens and align them with textual event semantics.

Response sampling. Given an observed video input $Q^{(3)}$, the model first generates imagined discrete visual tokens and then produces the final textual prediction. For each input, we sample a group of G responses from the old policy:

$$\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid Q^{(3)}), \tag{13}$$

where each response y_i contains both the generated reasoning trajectory and the final answer.

Reward computation. Each response receives a reward r_i composed of an accuracy reward and a format reward. The accuracy reward evaluates whether the parsed final prediction matches the recorded future event. For closed-set VEP, we parse the final option from the response and compare it with the ground-truth option. Responses with invalid, missing, or unparsable final answers receive no accuracy reward. The format reward encourages the model to follow the required output structure, including producing the imagined visual-token sequence and the final textual answer in the specified format. The final reward is computed as a weighted sum of the accuracy reward and the format reward.

Group-relative advantage. Following GRPO, we normalize rewards within each sampled group to compute the group-relative advantage:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^{G})}{\text{std}(\{r_j\}_{j=1}^{G}) + \delta}, \tag{14}$$

where δ is a small constant for numerical stability.

Importance-sampling ratio. To account for the distribution shift between the sampling policy and the updated policy, we use the token-level importance-sampling ratio:

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid y_{i,<t}, Q^{(3)})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid y_{i,<t}, Q^{(3)})}. \tag{15}$$

Clipped GRPO objective. The clipped GRPO objective is defined as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{Q^{(3)},\{y_i\}_{i=1}^{G}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \Big(\rho_{i,t}(\theta)\hat{A}_i,\ \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_i \Big) \right], \tag{16}$$

where ϵ is the clipping coefficient.

Final training loss. We optimize the model with the following loss:

$$\mathcal{L}_3 = -\mathcal{J}_{\text{GRPO}}(\theta) + \beta\, \mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right), \tag{17}$$

where π_{ref} denotes the reference policy and β controls the strength of the KL penalty. During Stage 3, both the codebook and visual-token embeddings are frozen, while the remaining trainable MLLM parameters are updated. The frozen codebook supplies a fixed quantization reference; this does not imply that all MLLM hidden representations remain unchanged.

F Descriptive Analyses

Figure 5 visualizes projected token features. Apparent proximity in this projection is descriptive and does not establish semantic fidelity, causal mediation, or absence of shortcuts.

F.1 Comparison of Training Stability

Figure 6 reports training curves for IDS and the continuous latent variant (Yang et al., 2025), which combines cosine feature supervision with answer-level cross-entropy. The variants use the same backbone and training data, but the continuous implementation supports only batch size 1 and has denser logging. IDS has smoother curves in the reported runs; differences in batch size and measurement frequency prevent attributing this observation solely to discretization. Moreover, opposite changes in the two logged losses are not a direct measurement of conflicting gradients. We therefore use these curves to describe the observed runs, not to establish a general optimization advantage. Appendix G defines the statistics and their interpretation.

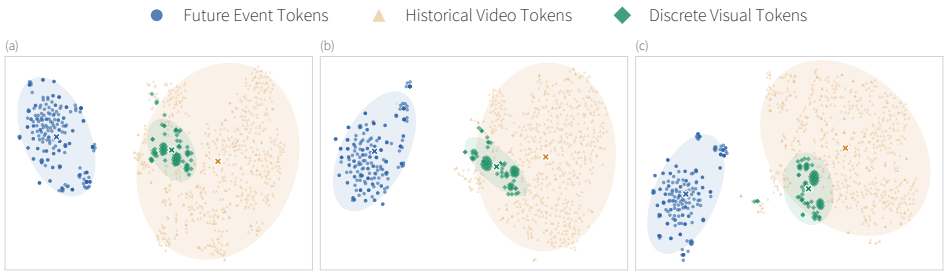


Figure 5: Token-level feature distribution visualization of the IDS reasoning process. Future Event Tokens denote text tokens corresponding to the recorded future event, Historical Video Tokens denote video tokens from the historical video, and Discrete Visual Tokens denote discrete visual tokens used for imagination during reasoning.

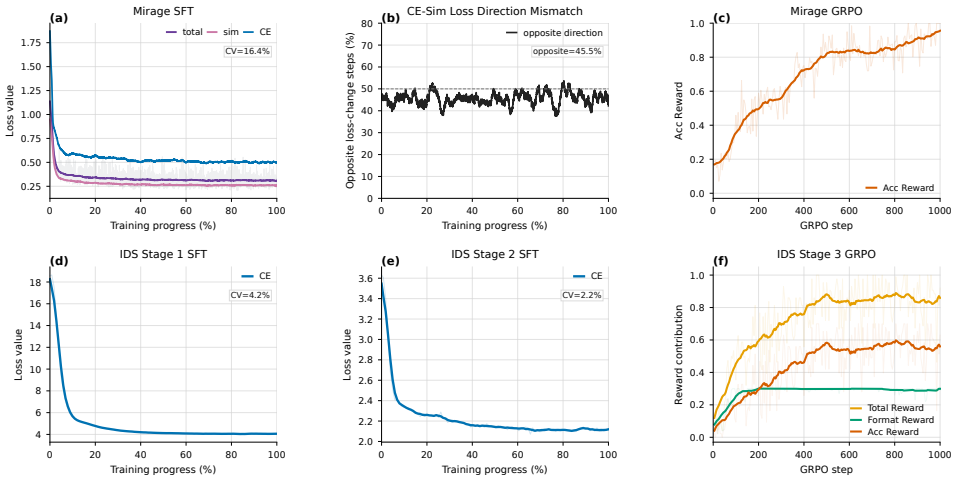


Figure 6: Training dynamics of discrete/continuous-space visual reasoning. CV measures relative loss fluctuation in the selected interval; opposite-direction steps measure disagreement in the signs of adjacent CE and similarity-loss changes. These statistics do not measure gradient conflict. See Appendix G for details.

G Training Stability Metrics

This section provides the definitions of the training-stability statistics reported in Figure 6. We use these statistics to characterize the relative fluctuation of the loss curves and the directions of adjacent logged loss changes.

Coefficient of Variation. The coefficient of variation (CV) measures the relative fluctuation of a loss curve within the stable training interval. Given a loss sequence $\{\ell_t\}_{t\in\mathcal{T}}$ over the selected interval $\mathcal{T}$, CV is computed as:

$$\mathrm{CV} = \frac{\mathrm{Std}\left(\{\ell_t\}_{t\in\mathcal{T}}\right)}{\mathrm{Mean}\left(\{\ell_t\}_{t\in\mathcal{T}}\right)} \times 100\%. \tag{18}$$

A larger CV indicates stronger relative oscillation, while a smaller CV suggests a smoother and more stable training curve. For example, $\mathrm{CV} = 16.4\%$ means that the standard deviation of the loss within the selected stable interval is approximately 16.4% of its mean.

CE-Sim direction mismatch. For continuous latent-space reasoning, the training objective contains both the autoregressive cross-entropy loss and the similarity loss. To measure whether these two losses locally evolve in consistent directions, we compute their step-wise changes:

$$\Delta\mathrm{CE}_t = \mathrm{CE}_t - \mathrm{CE}_{t-1}, \qquad \Delta\mathrm{Sim}_t = \mathrm{Sim}_t - \mathrm{Sim}_{t-1}. \tag{19}$$

A direction mismatch occurs when the two losses change in opposite directions:

$$m_t = \begin{cases} 1, & \Delta\mathrm{CE}_t \cdot \Delta\mathrm{Sim}_t < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

We then smooth this binary mismatch sequence with a rolling window of 501 steps:

$$\mathrm{Mismatch}_t = \frac{1}{501} \sum_{k=t-250}^{t+250} m_k \times 100\%. \tag{21}$$

Thus, for steps with a complete window, the value represents the percentage of neighboring steps in which the CE loss and similarity loss change in opposite directions. For instance, a value of 45% indicates that around the current step, approximately 45% of adjacent updates show opposite CE-Sim loss changes.

Interpretation. The dashed 50% line is a descriptive reference, not a calibrated statistical null. The overall mismatch ratio is 45.5%: adjacent logged losses change in opposite directions in that fraction of steps. Such changes depend on sampled minibatches, training trends, and logging frequency. They do not measure the inner product of the loss gradients and therefore do not establish objective conflict or explain a difference in optimization stability. Differences in batch size further limit comparisons between the reported curves.

H Effect of Option-B Reward Coefficient

Table 6 studies how different Option-B reward coefficients affect Qwen2.5-VL-7B.

Table 6: Ablation of the Option-B reward coefficient on Qwen2.5-VL-7B. The best results are highlighted in **bold**.

Option-B Reward	SR@A↑	SR@AB↑	SR@C↓	SR@D↓	SR@E↓
0.0	**72.78**	**76.62**	13.41	9.41	**0.57**
0.3	63.67	74.77	**9.80**	**8.07**	7.37
0.5	50.50	67.44	17.54	9.27	5.75
0.7	43.94	60.97	16.27	13.89	8.87

Setting $\eta = 0$ achieves the best SR@A and SR@AB, showing no improvement from the tested positive coefficients under this reward configuration. A small coefficient, such as $\eta = 0.3$, reduces SR@C and SR@D, but it also lowers SR@A and increases SR@E. The larger tested coefficients also reduce both primary metrics relative to zero. This result does not establish that rewarding alternative futures is generally ineffective. Therefore, we set the Option-B reward coefficient to 0 in the main experiments.

I PROMPT TEMPLATES

The following templates document the three training stages, open-set evaluation, and benchmark construction. The Stage 1 and Stage 2 prompts do not include answer candidates.

STAGE 1: FUTURE VISUAL-TOKEN PREDICTION

Stage 1: Prompt Template

System:

You are a professional analytical video interpreter. I will show you a video. Please first imagine the upcoming content based on the video, and then predict the most likely events that will happen.

User:

<video>Based on the video content, predict the most likely event to occur next.
To help your reasoning, first envision what might happen immediately after the video ends.
Output format: output your imagination within <Imagine></Imagine> tags.

STAGE 2: EVENT-DESCRIPTION GROUNDING

Stage 2: Prompt Template

System:

You are a professional analytical video interpreter. I will show you a video. Please first imagine the upcoming content based on the video, and then predict the most likely events that will happen.

User:

<video>Based on the video content, predict the most likely event to occur next.
To help your reasoning, first envision what might happen immediately after the video ends. Then, based on that, describe the future events you imagined.\n
Output format: first output your imagination within <Imagine></Imagine> tags, then enclose your thinking process within the <think></think> tags.

STAGE 3: REINFORCEMENT LEARNING

Stage 3: Prompt Template

System:

You are a professional analytical video interpreter. I will show you a video. Please first imagine the upcoming content based on the video, and then predict the most likely events that will happen.

User:

<video>Based on the video content, select the most likely event to occur next from the given options.
Options: \nXXXXXX\n
To help your reasoning, first envision what might happen immediately after the video ends. Then, based on that, describe the future events you imagined.\n
Output format: first output your imagination within <Imagine></Imagine> tags, then enclose your thinking process within the <think></think> tags, finally, output the option you think is most likely to happen in the <answer></answer> tag.

OPEN-SET EVALUATION

Model as Judge Prompt Template

You are a strict visual-evaluation assistant. Given a set of frames and N candidate model responses numbered 1..N, score how well each response matches the visual evidence of the recorded future video.
Score each sentence with an integer from 0 to 100 (100 = exact match, 0 = completely false). Also give a one-line justification (max 20 words) for each score.

IMPORTANT: Return ONLY a single JSON object as the entire assistant reply, with the exact keys:
{"scores": {"1": 90, ...}, "justifications": {"1": "short reason", ...}}
Do not include any extra text, commentary, or markup. Ensure scores are integers and justifications are brief.

BENCHMARK CONSTRUCTION

Benchmark Construction Prompt Template

Future Event
You are an assistant specialized in generating benchmark data for video event prediction. Your task is to create five event description options based on two consecutive video clips.
The first video clip describes a series of video events.
The second video clip is the future event to be predicted.
Task 1: Based on the logical correlation between two videos, output the difficulty of inferring the events of the second video from the first video
Task 2: Please generate Options A-E following the instructions below. Each option must be one sentence, written in clear and natural language.
Task 3: Generate the reasoning process that leads to the correct option A from the first video clip.
Task 4: Describe the difficulty of inferring the events of the second video based on the content of the first video. Easy: if the events in the second video follow naturally from the observed segment with clear visual cues; Middle: if the events in the second video have not occurred in the first video, but the answer can still be inferred through reasoning; Hard: if the events in the second video have changed significantly compared to the first video, such as the appearance of entirely new tasks or major scene changes.

General Rules
- Do not invent video content that does not exist in the video clip or the textual future event.
- Only Option A should accurately describe the future event.
- Options B-E must be incorrect in different ways, following their respective definitions.
- Ensure all descriptions are grammatically correct.

Option Definitions
** Option A **(Ground Truth Future Event)
Accurately summarize the main event that happens in the second video clip.
It must directly reflect the actual future event content.
** Option B ** (Next Event Inferred From Video Clip)
Describe a plausible next event that could reasonably happen after the video clip,
based only on information from the first video clip.
It must:
be obviously different from Option A
have a logical connection with the first video clip
not duplicate or continue the future event happened in the second video clip
** Option C ** (Happened Video Event)
Select an event that actually occurs within the first video clip
This must be an event fully contained within the first video clip, and obviously different from Option A.
** Option D ** (Subject-Modified Version of Option A)
Rewrite the event in Option A by changing the subject to another plausible entity
that appears in the video clip or the future event context.
The sentence must remain logical and consistent with common-sense knowledge.
Do not introduce impossible or unrealistic actions.
** Option E ** (Same Subject and Object as Option A, Unrelated Event)
Use the same subject and object as in Option A.
Create a new event that is:
NOT related to the future event
NOT a continuation, cause, or effect of Option A
The event must still be a sensible, ordinary real-world action.

Output Format
Please output your results in this exact structure:
difficulty:[easy,middle,hard]
Option A: [one-sentence description]
Option B: [one-sentence description]
Option C: [one-sentence description]
Option D: [one-sentence description]
Option E: [one-sentence description]
Reasoning: [reasoning process]
Ensure you follow the exact formatting above.