

# SEE THE FOREST FOR THE TREES: Loosely Speculative Decoding via Visual-Semantic Guidance for Efficient Inference of Video LLMs

Yicheng Ji<sup>1,2\*</sup>, Jun Zhang<sup>1,2\*</sup>, Jinpeng Chen<sup>3</sup>, Cong Wang<sup>4</sup>,  
Lidan Shou<sup>1,2</sup>, Gang Chen<sup>1</sup>, Huan Li<sup>1,2</sup>✉

<sup>1</sup>The State Key Laboratory of Blockchain and Data Security, Zhejiang University

<sup>2</sup>Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security

<sup>3</sup>School of Computer Science, Beijing University of Posts and Telecommunications

<sup>4</sup>School of Control Science and Engineering, Zhejiang University

{jiyicheng.cs, zj.cs, cwang85, should, cg, lihuan.cs}@zju.edu.cn, jpchen@bupt.edu.cn

## Abstract

Video Large Language Models (Video-LLMs) excel in video understanding but suffer from high inference latency during autoregressive generation. Speculative Decoding (SD) mitigates this by applying a draft-and-verify paradigm, yet existing methods are constrained by rigid *exact-match* rules, severely limiting the acceleration potential. To bridge this gap, we propose LVSPEC, the first training-free loosely SD framework tailored for Video-LLMs. Grounded in the insight that generation is governed by sparse visual-relevant anchors (mandating strictness) amidst abundant visual-irrelevant fillers (permitting loose verification), LVSPEC employs a lightweight visual-relevant token identification scheme to accurately pinpoint the former. To further maximize acceptance, we augment this with a position-shift tolerant mechanism that effectively salvages positionally mismatched but semantically equivalent tokens. Experiments demonstrate that LVSPEC achieves high fidelity and speed: it preserves **>99.8%** of target performance while accelerating Qwen2.5-VL-32B by **2.70×** and LLaVA-OneVision-72B by **2.94×**. Notably, it boosts the mean accepted length and speedup ratio by **136%** and **35%** compared to SOTA training-free SD methods for Video-LLMs. Code is at <https://github.com/zju-jiyicheng/LVSpec>.

## 1 Introduction

Video large language models (Video-LLMs) (Liu et al., 2023; Bai et al., 2023; Liu et al., 2024; Lin et al., 2024; Li et al., 2024b; Bai et al., 2025; Li et al., 2025a; An et al., 2025) show strong performance in video understanding tasks such as video captioning and video question answering. However, inference latency remains a key bottleneck due to autoregressive backbones, hindering practical deployment. For instance, to process a 100-second high-resolution video at 32 FPS, Qwen2.5-VL (Bai et al., 2025) by default encodes over one million

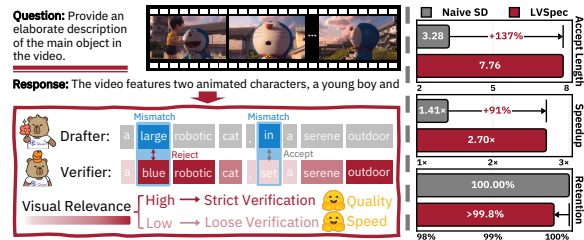


Figure 1: LVSPEC perform strict verification for visual-relevant tokens and loose verification for visual-irrelevant ones, boosting efficiency while preserving performance.

visual tokens. The long sequence of tokens, together with the full model parameters, requires autoregressive access during decoding, leading to a memory-bound bottleneck and increasing end-to-end latency.

To achieve lossless decoding-time acceleration, *Speculative Decoding (SD)* offers a promising alternative. It leverages a lightweight *draft model* to propose multiple draft tokens, which are then verified in parallel by a *target model*. Recent works (Ji et al., 2025; Kang et al., 2025; Ganesan et al., 2025; Huo et al., 2025; Bajpai and Hanawal, 2025; Xie et al., 2025; Huang et al., 2025) have explored SD for large vision-language models, and SpecVLM (Ji et al., 2025) has pioneered the application of SD to Video-LLMs in a training-free manner. However, existing SD frameworks are fundamentally constrained by their *exact-match* rule: a draft token is accepted only if it is identical to the target model’s generation. Such a strictly SD scheme forces the unfavorable rejection of many *semantically aligned* tokens, leading to limited mean accepted lengths and speedup ratios.

For large language models (LLMs), loosely speculative decoding has evolved from costly classifiers (Bachmann et al., 2025) to training-free heuristics like FLY (Li et al., 2025b). Specifically, FLY accepts a mismatched token as a semantically equivalent alternative only when model uncertainty is high (gauged by entropy) and the divergence proves transient within a deferred window. Con-

\*Equal contribution. ✉Corresponding author.

sequently, this approach achieves highly generalizable acceleration with minimal overhead. However, naively adapting these advanced loosely SD tailored for LLMs to Video-LLMs suffers from a critical “visual blindness” limitation. By relying solely on linguistic priors, such as output entropy, these methods apply an indiscriminate verification standard, treating visually critical descriptors and trivial syntactic fillers with the same logic. We posit that for Video-LLMs, generation quality is defined not by rote lexical matching, but by the fidelity of articulating key visual information. This intuition raises a pivotal question:

*Are all decoded tokens equally critical for visual understanding, thus mandating uniform verification rigor?*

Our answer is negative. We assert that visual grounding must dictate verification strictness, as shown in Figure 1. On one hand, tokens strongly tied to visual content (e.g., “blue”) act as the anchors of understanding; treating them loosely (as text methods might do for high-entropy synonyms) risks factual hallucinations. Therefore, they mandate strict validation. On the other hand, visually agnostic tokens (e.g., “set”) contribute minimally to fidelity; maintaining strict matching here (as text methods do for low-entropy non-synonyms) is computationally wasteful. Consequently, they permit adaptive and relaxed matching. Driven by this insight, we introduce LVSPEC, a novel loosely SD framework tailored for Video-LLMs that utilizes visual-semantic guidance.

LVSPEC introduces a **visual-relevant text token identification scheme** in Section 3.1 that computes the visual similarity of text tokens at each decoding step, which accurately identifies the key text tokens that carry the richest visual cues. Subsequently, we adopt a **visual-semantic guided loose verification** in Section 3.2 that selectively token matching constraint based on the above quantified visual relevance. Moreover, we introduce a **position shift-tolerant mechanism** in Section 3.3 that relaxes acceptance for tokens that appear in the draft sequence but mismatch due to positional shifts, thereby further improving speedup without reducing the conveyed visual information.

To summarize, our main contributions are:

- (1) *Novel Discovery and Theoretical Potential:* We demystify the dual nature of visual dependence: generation is governed by  $\leq 20\%$  sparse yet critical visual-relevant anchors amidst over 60% visual-irrelevant linguistic fillers. Capitalizing on this, we provide a rigorous theoretical analysis to quantify the acceleration potential, guaranteeing strict verification for anchors while relaxing the rest.

- (2) *Visual-Semantic Guided Loosely SD:* To unleash this potential, we propose LVSPEC, the first loosely SD tailored for Video-LLMs. By employing a lightweight visual-relevant token identification and loose verification scheme, augmented by a position-shift tolerant mechanism, LVSPEC shatters the rigid “exact-match” barrier, effectively relaxing unnecessary strict verification via text-vision relevance.
- (3) *Extensive Experiments and Analysis:* Comprehensive evaluations across four video understanding benchmarks show that LVSPEC is both high-fidelity and rapid: it preserves **>99.8%** of target performance while accelerating Qwen2.5-VL-32B by **2.70 $\times$**  and LLaVA-OneVision-72B by **2.94 $\times$** . Notably, it boosts the mean accepted length and speedup ratio by a remarkable **136%** and **35%** compared to SOTA training-free SD for Video-LLMs, validating our visual-centric design.

## 2 Preliminary Study

In this section, we investigate the decoding dynamics of Video-LLMs to motivate our proposed approach. We begin by presenting empirical evidence in Section 2.1, which reveals the **dual nature** of visual semantic dependence: it is **sparse in distribution yet decisive for semantic grounding**. Building on this observation, we provide a rigorous theoretical analysis in Section 2.2 to **quantify the theoretical acceleration potential** of exploiting this sparsity, establishing the mathematical foundation for breaking the bottleneck of strictly SD.

### 2.1 Sparse yet Critical Visual Semantics

To investigate the underlying mechanism of visual understanding in Video-LLMs, we conducted an **oracle analysis**. We term this an “oracle” study because it relies on retrospective classification by a superior model (e.g., GPT-4o) using full context information that is theoretically inaccessible during real-time decoding. We utilized four advanced production-grade models (GPT-4o, Gemini-3-Pro, DeepSeek-V3, and Qwen3-Max) on the Video Detail Caption (VDC) (Chai et al., 2025) benchmark to address two fundamental questions regarding token distribution and importance.

**Question 1:** *Are visual semantics uniformly distributed across the generated sequence or are they concentrated in specific units?*

To address this, we collect generation responses of Qwen2.5-VL-32B on VDC and prompt the four production-grade models to categorize each generated token based on the complete sentence context. We established a taxonomy where “Visual-Relevant” refers to tokens directly corresponding to

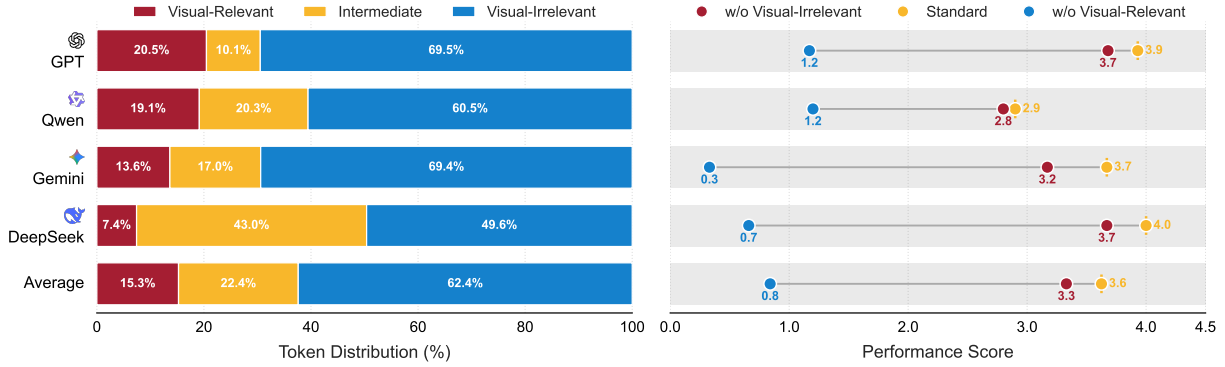


Figure 2: **Left:** (a) The distribution of visual-relevant and visual-irrelevant tokens. **Right:** (b) LLMs evaluation on Video Detail Caption benchmark. Visual-relevant tokens dominate the output quality.

visible scene elements, “Visual-Irrelevant” denotes function or grammatical words, abstract terms, conjunctions, and “Intermediate” covers other terms<sup>1</sup>. As shown in Figure 2, the distribution of these categories reveals a consistent pattern:

**Observation 1 (Sparsity):** *Only a small fraction of generated tokens are directly tied to visual elements, while the vast majority consist of functional, grammatical, or abstract fillers that are either completely irrelevant or only loosely associated with the visual content.*

Having established the distributional sparsity of visual tokens, we proceed to examine their contribution to the overall generation quality.

**Question 2:** *Are all decoded tokens equally critical for visual understanding, or does their impact on quality vary significantly?*

We investigate this by performing a targeted ablation across all four models, selectively pruning the top-50 most versus least visually relevant tokens from the normalized outputs. The subsequent evaluation (shown in Figure 2 (b)) demonstrates a sharp contrast: removing visually relevant tokens leads to a substantial collapse in generation quality, whereas removing high-frequency irrelevant tokens results in no noticeable performance drop. This leads to our second observation:

**Observation 2 (Criticality):** *The semantic integrity of the generation relies heavily on the presence of visually relevant tokens; functional and irrelevant tokens, despite their high frequency, contribute negligibly to the core visual understanding.*

We distill these empirical observations into the following hypothesis:

**Visual Anchor Sparsity Hypothesis:** Generation quality is governed by a sparse set of “anchor” tokens, while dense irrelevant “filler” tokens carry minimal information density.

<sup>1</sup>Figure 11 in Appendix H shows the prompt template.

This insight motivates our verification strategy: enforcing strict verification on anchors while relaxing verification for the remaining fillers aggressively<sup>2</sup>.

## 2.2 Theoretical Benefits of Loose Verification

We first formalize the speculative decoding (SD) process. Given a target model (verifier)  $\mathcal{M}_t$  and a draft model (drafter)  $\mathcal{M}_d$ , in each decoding step  $s$ , the draft model  $\mathcal{M}_d$  autoregressively produces  $K$  draft tokens  $\{\hat{y}_k^s\}_{k=1}^K$ , which are then verified by the target model in parallel:

$$\{y_k^s\}_{k=1}^K = \mathcal{M}_t([y^0 \cdots y^{s-1}, \{\hat{y}_k^s\}_{k=1}^K]), \quad (1)$$

where  $\{y_k^s\}_{k=1}^K$  denotes the verified sequence and  $y^0 \cdots y^{s-1}$  represent the context history. Following Li et al. (2025b), we define the verification strategy  $\nabla$  to determine whether the draft sequence matches the verified sequence at each position (1 if matched, else 0). The accepted length  $\tau_s^\nabla$  (a random variable) for the current step  $s$  is defined using the longest common prefix logic:

$$\tau_s^\nabla := \sum_{k=1}^K \left( \prod_{j=1}^k \nabla(\hat{y}_j^s, y_j^s) \right). \quad (2)$$

For notational brevity, we assume the decoding process is statistically stationary across steps, rendering the expected accepted length invariant to  $s$ <sup>3</sup>. Thus, we drop the index and denote the expectation as  $\mathbb{E}[\tau^\nabla]$ . Let  $T_t$  and  $T_d$  be the latency of  $\mathcal{M}_t$  and  $\mathcal{M}_d$  per step, respectively. The overall speedup ratio is defined as:

$$\text{speedup} := \frac{\mathbb{E}[\tau^\nabla] \cdot T_t}{K \cdot T_d + T_t^K}. \quad (3)$$

Here  $T_t^K$  denotes the latency for  $\mathcal{M}_t$  to verify  $K$  draft tokens in parallel. Based on Equation (3), we analyze the behavior of  $\mathbb{E}[\tau^\nabla]$  under different

<sup>2</sup>See Figure 12 in Appendix H for a detailed case analysis.

<sup>3</sup>This simplification does not alter our findings.

verification strategies. Specifically, we **contrast the theoretical limits** of standard strict verification ( $\mathbb{E}[\tau_{\text{strict}}^{\nabla}]$ ) **against the scaling capabilities** of loose verification, quantified by the ratio  $\frac{\mathbb{E}[\tau_{\text{loose}}^{\nabla}]}{\mathbb{E}[\tau_{\text{strict}}^{\nabla}]}$ .

**Proposition 2.1** (Acceptance Bound of Strict Verification). *Let  $\alpha$  denote the draft alignment accuracy and  $\epsilon := 1 - \alpha$  be the failure rate. The expected accepted length for strict verification is theoretically bounded by the inverse of the failure rate (proof in Appendix A.2).<sup>4</sup>*

$$\mathbb{E}[\tau_{\text{strict}}^{\nabla}] = \sum_{k=1}^K \alpha^k \approx \frac{\alpha}{1 - \alpha} < \frac{1}{\epsilon}. \quad (4)$$

The upper bound established in Proposition 2.1 exposes an intrinsic **efficiency bottleneck**: the speedup is shackled by the draft model’s raw failure rate  $\epsilon$ . Existing strictly SD methods strive to improve the mean accepted length  $\mathbb{E}[\tau^{\nabla}]$  *indirectly*, either by enhancing the alignment of the draft model through curated training or involving heavy draft tree structures. In contrast, loosely SD approaches can **directly and linearly scale up**  $\mathbb{E}[\tau^{\nabla}]$  without increasing any draft overhead. We formalize this capability in the following theorem.

**Theorem 2.2** (Scaling Law under Loose Verification). *Let the token sequence be partitioned into sparse yet critical visual-relevant tokens  $\mathcal{V}$  and dense yet redundant visual-irrelevant tokens  $\mathcal{V}'$  with a visual density  $\rho := |\mathcal{V}|/K$ , where  $\rho \ll 1$ . By **exploiting this sparsity**, the proposed loose verification mechanism achieves a scaling ratio  $\gamma$  inversely proportional to the visual density (proof in Appendix A.3):*

$$\frac{\mathbb{E}[\tau_{\text{loose}}^{\nabla}]}{\mathbb{E}[\tau_{\text{strict}}^{\nabla}]} \approx \frac{1}{\rho}. \quad (5)$$

**Remark 2.3** (Diluting the Failure Rate). Theorem 2.2 offers a rigorous guarantee: our mechanism mathematically **dilutes the raw failure rate  $\epsilon$  into a significantly lower effective failure rate  $\rho\epsilon$** . By effectively “bypassing” the strict geometric decay on visual-irrelevant tokens, we enable SD to surpass the theoretical ceiling imposed by the draft model alone. Thus, such a loose verification exhibits high tolerance to draft errors, significantly expanding the feasible region for acceleration even when the base alignment  $\alpha$  is suboptimal.

Driven by the empirical observations and theoretical insights, we present LVSPEC, a novel loosely speculative decoding for Video-LLMs. LVSPEC operationalizes the concept of loose verification by leveraging visual semantics, thereby translating the

<sup>4</sup>Based on the standard i.i.d. assumption in Appendix A.1.

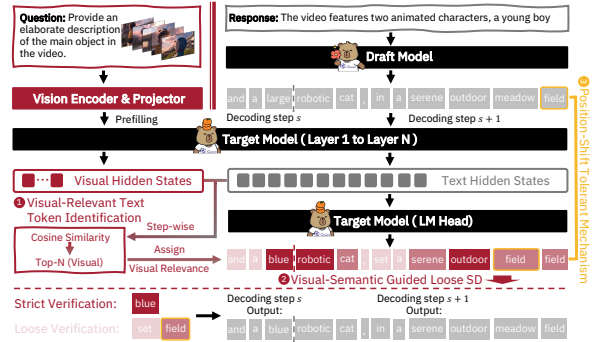


Figure 3: Overview of LVSPEC.

forementioned theoretical potential into practical wall-clock speedup.

### 3 LVSPEC

As depicted in Figure 3, LVSPEC first identifies visual-relevant tokens at each decoding step (Section 3.1), then strictly verifies relevant ones while loosely verifying irrelevant ones (Section 3.2). Additionally, we design a position shift-tolerant mechanism to accept mismatched tokens appearing in a near span of the draft sequence (Section 3.3).

#### 3.1 Visual-Relevant Text Token Identification

In Section 2.1, we empirically demonstrate that visual understanding tasks exhibit a pronounced disparity in sensitivity between visual-relevant and visual-irrelevant tokens in the output. However, to exploit this discrepancy for accelerating inference, we require a metric that can quantify visual relevance *during* the decoding stage.

In prior work (Zhang et al., 2025d), the cosine similarity between vision embeddings and text embeddings is used to measure their relevance. Since the vision embeddings are projected into the same textual embedding space via the vision projector, the distance between these two representations can effectively reflect their association in the semantic space within the LLM backbone. Leveraging such cross-modal similarity, prior work (Zhang et al., 2025d) focuses on text-guided visual token pruning, whereas LVSPEC takes the opposite perspective: we apply a novel vision-guided approach to facilitate text generation.

Specifically, we decompose  $\mathcal{M}_t$  into the final-layer representation function  $\tilde{\mathcal{M}}_t(\cdot)$  and the language modeling head  $\text{lm\_head}(\cdot)$  as follows:

$$\mathcal{M}_t(\cdot) \rightarrow \text{lm\_head}(\tilde{\mathcal{M}}_t(\cdot)), \quad (6)$$

where  $\tilde{\mathcal{M}}_t(\cdot)$  denotes the output hidden states of the last layer of  $\mathcal{M}_t$ , before the language model head. We use  $\tilde{\mathcal{M}}_t(\cdot)$  for subsequent computations because (i) the Transformer’s final-layer hidden

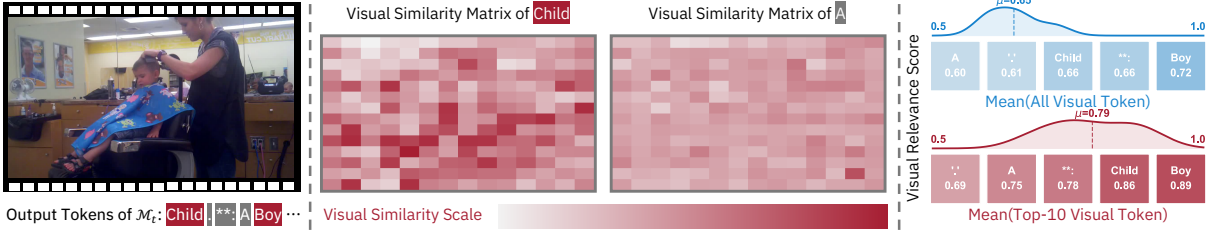


Figure 4: **Left:** (a) Decoded tokens from the target model  $\mathcal{M}_t$ . **Middle:** (b) Visualization of the visual similarity matrix for the decoded tokens. **Right:** (c) Distribution of visual relevance score w/o and w/ Top-N (N=10) selection.

states provide a strong feature representation, consistent with the design choice in EAGLE (Li et al., 2024d), and (ii) the sequence produced by  $\mathcal{M}_t$  determines the final output, rather than  $\mathcal{M}_d$ .

Then, for a given input video and prompt, let  $E_V \in \mathbb{R}^{l_v \times d}$  be the input embedding after the vision encoder and projector, where  $l_v$  is the visual sequence length and  $d$  is the hidden dimension of the target model  $\mathcal{M}_t$ . At each decoding step  $s$ , the draft model  $\mathcal{M}_d$  generates  $K$  draft tokens, yielding the corresponding draft text embedding  $E_D^s \in \mathbb{R}^{K \times d}$ . We extract the hidden states  $H_V$  (during prefilling) and  $H_D^s$  (during decoding) as:

$$\begin{aligned} H_V &= \tilde{\mathcal{M}}_t(E_V) \in \mathbb{R}^{l_v \times d}, \\ H_D^s &= \tilde{\mathcal{M}}_t(E_D^s) \in \mathbb{R}^{K \times d}. \end{aligned} \quad (7)$$

In each decoding step,  $H_V$  and  $H_D^s$  are used for similarity computation. Normally, visual cues consist of both salient subjects and abundant background elements. As visualized in Figure 4 (a,b), we find that visual-relevant tokens associated with the main subject (e.g., “Child”) exhibit a much sharper similarity profile: they align strongly with the salient subject cues while showing low similarity to the background. In contrast, less relevant tokens (e.g., “A”) tend to have a more uniform similarity distribution across all background cues. If the full  $H_V$  containing all video cues is used as the reference, the similarity of visual-relevant tokens can be diluted by abundant background cues, making them difficult to distinguish from visual-irrelevant ones. Therefore, we compute the cosine similarity between the text hidden states and the visual hidden states, and take the average of the Top- $N$  largest similarities along the visual dimension as the final **visual relevance** metric  $\mathcal{A}_s$  for each text token in the decoding step  $s$ , denoted as:

$$\begin{aligned} \mathcal{A}_s &= \frac{1}{N} \sum_{n=1}^N \text{TopN}(\cos(H_D^s, H_V), N), \\ \text{where } \cos(H_D^s, H_V) &= \frac{H_D^s \cdot H_V}{\|H_D^s\| \|H_V\|}. \end{aligned} \quad (8)$$

Here,  $N$  is a hyper-parameter controlling the number of critical visual tokens that provide guidance.

This operation effectively improves the discriminability of the visual relevance (see Figure 4 (c)), with minimal computation cost (see Figure 10).

### 3.2 Visual-Semantic Guided Loosely SD

Given the visual similarity of text tokens in each decoding step  $\mathcal{A}_s$ , we leverage it as guidance to perform loosely speculative decoding. Specifically, we adopt strict verification for the tokens with high visual relevance, while allowing the acceptance of visual-irrelevant tokens. Let  $\lambda \in [0, 1]$  be a loose parameter to control the percentage of visual-irrelevant tokens that are loosened. Then in each decoding step, we consider the  $\lambda K$  tokens with the *lowest* visual relevance scores as the practical instantiation of  $\mathcal{V}'$ , aligning with the redundancy assumption in Theorem 2.2. Then the index set of  $\mathcal{V}'$  can be denoted as:

$$\mathcal{I}'_s = \text{argsort}_{\uparrow}(\mathcal{A}_s)_{0:\lambda K}, \quad (9)$$

Finally, we define the verification strategy as  $\nabla_{\text{LVSPEC}}$  to decide whether the draft tokens match the target model’s verified tokens at each position:

$$\nabla_{\text{LVSPEC}}(y_i^s, \hat{y}_i^s) = \begin{cases} 1, & \text{if } y_i^s \text{ matches } \hat{y}_i^s, \\ 1, & \text{else if } i \in \mathcal{I}'_s, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

By choosing a larger  $\lambda$ , we can apply a more aggressive relaxation to pursue higher speedups. Conversely, for tasks that require a more conservative behavior, we can use a smaller  $\lambda$ , eventually recovering the standard exact-match scheme as  $\lambda$  decreases. This mechanism allows the level of relaxation to be adaptively tuned according to task characteristics and model behaviors.

Notably, the verification above is based on the local visual relevance among text tokens at each decoding step. This design is chosen because hidden state values of nearby text tokens are more directly comparable and are less susceptible to absolute drifts induced by the growing generation context.

### 3.3 Position Shift-Tolerant Mechanism

As illustrated in Figure 5, we observe that in video understanding outputs, occasional mismatches

Case 1	Output Tokens of $\mathcal{M}_d$ :	sail boat navigating the open sea
	Output Tokens of $\mathcal{M}_t$ :	sail boat navigating open open sea
Case 2	Output Tokens of $\mathcal{M}_d$ :	featuring a wooden deck and multiple sails
	Output Tokens of $\mathcal{M}_t$ :	featuring multiple wooden deck and multiple sails

Figure 5: Insight cases of PST. Key background colors denote **Mismatched Draft Token**, **Shifted Draft Token**, and **Verified Target Token**, respectively.

arise due to positional shifts. Specifically, such shifts are typically caused by the draft model introducing additional phrases (token “the” in *case 1*), or by differences in the description order between the draft and target models (token “multiple” in *case 2*). We argue that in such cases, the target model’s intended output is covered by the draft model’s output distribution. In Figure 5, the draft output is actually aligned with that of the target model and relaxing the currently mismatched token may not lead to a loss of visual information in subsequent generations. Based on this finding, we further incorporate a position shift-tolerant mechanism (PST), a bonus component for such cases. If the currently mismatched verified token appears within a nearby span of the draft sequence, we treat the mismatch as position shift-induced and accept it. Specifically, we set the draft length  $K$  at each decoding step as the tolerable shift range, and the verification strategy  $\nabla_{\text{LVSPEC}}$  is reformulated as:

$$\nabla_{\text{LVSPEC}}(y_i^s, \hat{y}_i^s) = \begin{cases} 1, & \text{if } y_i^s \text{ matches } \hat{y}_i^s, \\ 1, & \text{else if } i \in \mathcal{I}'_s, \\ 1, & \text{else if } y_i^s \text{ is in } \{\hat{y}_k^s\}_{k=1}^K, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Finally, the accepted length is defined using the longest common prefix logic. Note that PST serves only as a supplementary strategy after visual-semantic guidance is applied, selectively enabled in minority cases depending on the output pattern<sup>5</sup>.

## 4 Experiment

**Tasks and Benchmarks.** We conduct experiments on four video captioning and video question answering tasks, including Video Detail Caption (VDC) (Chai et al., 2025), Video Detail Description (VDD) (LMMS-Lab, 2024), MovieChat (Song et al., 2024), and Video-MME (Fu et al., 2025), using LMMS-Eval (Li et al., 2024a) for evaluation. The performance metric is evaluated by LLM judges using the standard prompt template. Benchmark details are provided in Appendix B.

**Baselines.** We compare our method with both training-free lossless SD methods and loosely SD

methods. (i) **Lossless SD methods.** ① NAIVE SD directly uses an existing model as the draft model. ② SPECVLM (Ji et al., 2025) applies uniform video token pruning to the draft model, reserving 10% of the visual tokens to make it compact. (ii) **Loosely SD methods.** ③ FLY (Li et al., 2025b) consists of an entropy gate and a deferred window: a mismatched token is accepted when its entropy exceeds a threshold and all tokens within the subsequent window match. ④ FLY<sup>⊖</sup> uses only the entropy gate component of FLY, serving as a more loosened variant. Both NAIVE SD and SPECVLM adopt the best-performing draft tree configuration in the original paper, consistent with EAGLE (Li et al., 2024d). For loosely SD methods FLY<sup>⊖</sup>, FLY, and LVSPEC, we use the same number of draft tokens ( $K = 10$ ), forming a chain-like structure that also matches the design in FLY.

**Target and Draft Models.** We choose two representative Video-LLMs families: Qwen2.5-VL (Bai et al., 2025) and LLaVa-OneVision (Li et al., 2025a) as our target and draft models. Aligning with SPECVLM, we evaluate two speculative decoding settings. (i) **Standard-SD (Std.-SD):** using a smaller Video-LLM from the same model family. “Std.-SD-Qwen2.5-VL” and “Std.-SD-LLaVA-OV” denote settings that use Qwen2.5-VL-32B and LLaVA-OneVision-72B as the target models, and Qwen2.5-VL-7B and LLaVA-OneVision-7B with video token pruning applied as the corresponding draft models, respectively. (ii) **Self-SD:** using the original model and full video tokens as verifier, while using the original model with pruned video tokens as drafter. “Self-SD-Qwen2.5-VL” and “Self-SD-LLaVA-OV” adopt Qwen2.5-VL-7B and LLaVA-OneVision-7B, respectively. A study of additional draft model sizes is provided in Appendix D. We do not introduce additional training for the draft model, as our primary focus is to isolate how the verification strategy affects the mean accepted length  $\tau$  and speedup ratio.

**Implementation Details.** For all experiments, we set the loose parameter  $\lambda = 0.7$  and the number of critical visual tokens  $N = 10$ . We set the maximum generation length to 512 tokens and conduct all experiments on two NVIDIA H200 GPUs. More details of the implementation of FLY (Li et al., 2025b) and SPECVLM (Ji et al., 2025) are elaborated in Appendix C.

### 4.1 Main Result

We evaluate the **efficiency metric** including speedup ratio and the mean accepted token length  $\tau$ , and the **performance metric** including rating score and accuracy on video understanding tasks, as summarized in Table 1 and Figure 6.

<sup>5</sup>Potential improvement of LVSPEC is in Appendix I.

Model	Method	Video Detail Caption (VDC)			Video Detail Description (VDD)			$\tau$	MovieChat	
		$\tau$	Token/s	Speedup	$\tau$	Token/s	Speedup		Token/s	Speedup
Std.-SD Qwen2.5-VL	Autoregressive	–	7.95	1.00×	–	10.07	1.00×	–	9.81	1.00×
	NAIVE SD	3.28	11.17	1.41×	3.36	14.44	1.42×	3.19	13.38	1.36×
	SPECVLM	3.29	15.90	2.00×	3.31	17.96	1.76×	3.19	17.05	1.74×
	FLY <sup>⊖</sup>	6.81	18.50	2.32×	7.20	22.14	2.20×	6.74	20.76	2.12×
	FLY	4.37	13.60	1.71×	4.34	14.51	1.44×	3.86	13.01	1.33×
	LVSPEC (Ours)	<b>7.76</b>	<b>21.47</b>	<b>2.70×</b>	<b>7.68</b>	<b>23.58</b>	<b>2.34×</b>	<b>7.74</b>	<b>21.49</b>	<b>2.19×</b>
Self-SD Qwen2.5-VL	Autoregressive	–	20.60	1.00×	–	24.75	1.00×	–	25.51	1.00×
	NAIVE SD	4.43	18.21	0.88×	4.46	23.22	0.94×	4.46	21.86	0.86×
	SPECVLM	3.85	31.45	1.53×	3.89	35.00	1.41×	3.66	32.20	1.26×
	FLY <sup>⊖</sup>	6.86	32.24	1.56×	7.17	33.58	1.36×	5.98	28.23	1.11×
	FLY	6.53	28.16	1.37×	6.35	30.30	1.22×	5.44	25.82	1.01×
	LVSPEC (Ours)	<b>8.36</b>	<b>36.49</b>	<b>1.77×</b>	<b>8.64</b>	<b>39.32</b>	<b>1.59×</b>	<b>7.99</b>	<b>34.04</b>	<b>1.33×</b>
Std.-SD LLaVA-OV	Autoregressive	–	6.62	1.00×	–	6.00	1.00×	–	4.41	1.00×
	NAIVE SD	3.40	12.50	1.89×	3.72	11.74	1.96×	3.88	5.36	1.22×
	SPECVLM	3.37	15.75	2.38×	3.53	14.94	2.49×	3.77	8.45	1.92×
	FLY <sup>⊖</sup>	4.83	13.82	2.09×	4.93	11.90	1.98×	5.55	9.23	2.09×
	FLY	4.12	12.17	1.84×	4.58	11.50	1.92×	5.01	5.72	1.30×
	LVSPEC (Ours)	<b>7.34</b>	<b>19.44</b>	<b>2.94×</b>	<b>7.59</b>	<b>17.45</b>	<b>2.91×</b>	<b>8.10</b>	<b>12.16</b>	<b>2.75×</b>
Self-SD LLaVA-OV	Autoregressive	–	26.45	1.00×	–	17.66	1.00×	–	17.23	1.00×
	NAIVE SD	4.58	23.83	0.90×	4.59	15.74	0.89×	4.79	11.16	0.65×
	SPECVLM	3.91	35.46	1.34×	3.95	29.08	1.65×	4.17	<b>21.19</b>	<b>1.23×</b>
	FLY <sup>⊖</sup>	6.83	31.72	1.20×	6.82	27.01	1.45×	8.06	17.94	1.04×
	FLY	6.23	29.39	1.11×	6.20	25.57	1.45×	7.65	17.29	1.00×
	LVSPEC (Ours)	<b>8.46</b>	<b>36.40</b>	<b>1.38×</b>	<b>8.87</b>	<b>34.96</b>	<b>1.98×</b>	<b>8.77</b>	20.28	1.18×

Table 1: Speedup ratios and mean accepted length  $\tau$  on video captioning and video question answering tasks.

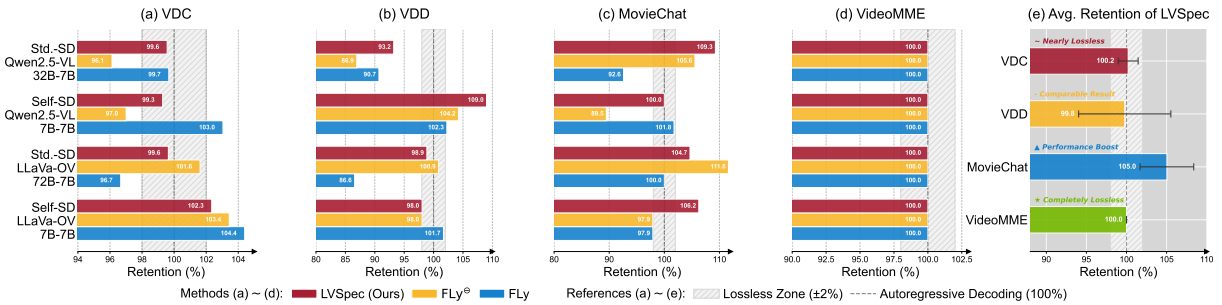


Figure 6: Performance retention of loosely SD methods. Table 7 in Appendix G shows the raw performance data.

**LVSPEC shows superior speedup and mean accepted length  $\tau$  across various model settings and video understanding tasks.** As shown in Table 1, LVSPEC achieves the highest speedups of 2.70 $\times$  and 2.94 $\times$  under the Std.-SD-Qwen2.5-VL and Std.-SD-LLaVA-OV settings, respectively. The mean accepted length  $\tau$  of LVSPEC in the Std.-SD setting ranges from 7.34 to 8.10, which is 2.15 $\times$  that of the lossless SD baseline SPECVLM and  $>$  1.46 $\times$  that of the loosely SD baseline FLY<sup>⊖</sup> and FLY. For the Self-SD-Qwen2.5-VL and Self-SD-LLaVA-OV settings, LVSPEC also achieves the highest mean accepted length  $\tau$  (8.87) and the highest speedup (1.98 $\times$ ). In addition, the observed improvements of LVSPEC with  $\lambda = 0.7$  closely matches the effect of  $\rho$  on the failure rate analyzed in Theorem 2.2 (see Appendix G for a detailed explanation). On tasks that require longer output such as VDC and VDD, LVSPEC tends to achieve higher speedups. On Self-SD LLaVA-OV setting and MovieChat task, LVSPEC attains a slightly lower speedup than SPECVLM. We attribute this to the already high  $\tau$  before relaxation in this setting, which leaves limited room for relaxation, as well as the fact that SPECVLM employs

a draft tree with more verification nodes (26) than LVSPEC (10). In Appendix E, we additionally report results of SPECVLM w/o draft trees as well as LVSPEC incorporating draft trees.

**LVSPEC achieves nearly lossless performance on video captioning and open-ended question answering tasks, and remains fully lossless on multiple-choice task compared with the target model’s original output.** As shown in Figure 6, LVSPEC achieves average accuracy retention ratios of 100.2%, 99.8%, 105.0%, and 100.0% on VDC, VDD, MovieChat, and Video-MME, respectively. In comparison, the retention ratios of the draft model on the four tasks are 94.7%, 95.2%, 105.9%, and 89.3% on average. Across all model settings and tasks, LVSPEC retains at least 93.2% of the original accuracy, outperforming FLY<sup>⊖</sup> (86.9%) and FLY (86.6%), and demonstrating better overall stability. While preserving performance, LVSPEC also achieves higher speedups than these baselines (1.58 $\times$  of FLY on VDC), indicating that its loosened criterion is more precise and effective. Notably, on MovieChat, LVSPEC even yields a 5.0% accuracy improvement. We attribute this to the higher sensitivity of short generation tasks to small

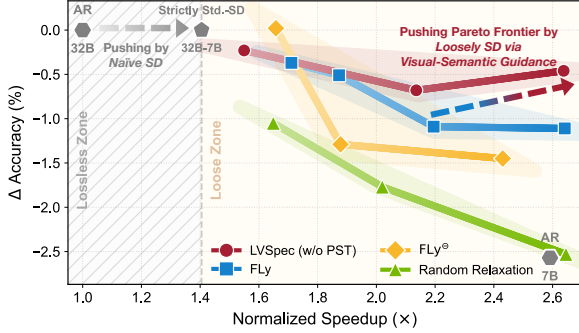


Figure 7: The Pareto frontier of accuracy and speedup for Video-LLM decoding methods under Qwen2.5-VL-32B/7B and VDC task.

Model	PST	$\tau$	Speedup	Retention (%)
Std.-SD Qwen2.5-VL	×	7.27	2.54×	98.5
Std.-SD LLaVa-OV	×	6.89	2.82×	99.1
Qwen2.5-VL	✓	7.76	2.70×	99.6
LLaVa-OV	✓	7.34	2.94×	99.8

Table 2: Ablation study on position shift-tolerant (PST).

perturbations, and that some existing small models can occasionally outperform the large model in certain settings (see Table 7 in Appendix G).

## 4.2 Ablation Study

We study the effect of the visual relevance metric and the effect of the PST. All experiments in this section are conducted under Qwen2.5-VL-32B/7B and VDC benchmark, using accuracy as metric.

**Effect of Visual-Semantic Guidance.** Here, we evaluate the loose criterion using only visual-semantic guidance (w/o PST,  $\lambda \in \{0.2, 0.5, 0.7\}$ ) against baselines including FLY<sup>⊖</sup> (Entropy  $\in \{0.1, 0.15, 0.175\}$ ), FLY ((Window, Entropy)  $\in \{(4, 0.1), (2, 0.1), (1, 0.1), (1, 0.05)\}$ ), and random relaxation with sampling probabilities ( $\{0.25, 0.5, 0.75\}$ ). As shown in Figure 7, LVSPEC effectively pushes the Pareto frontier by leveraging visual relevance to precisely loosen the verification criterion. Consistent with Theorem 2.2, LVSPEC achieves a superior accuracy-speedup trade-off compared to “visually blind” baselines, which fail to maintain effectiveness under higher relaxation levels. While lossless SD ensures quality, LVSPEC enables significant acceleration in the loose regime with only marginal quality costs.

**Effect of PST.** For the position shift-tolerant mechanism (PST), as shown in Table 2, incorporating PST boosts the speedup from 2.54× to 2.70× and  $\tau$  from 7.27 to 7.76 for Std.-SD-Qwen2.5-VL, with consistent gains for Std.-SD-LLaVa-OV. Notably, PST does not reduce accuracy, indicating

Hyperparameter	Val	$\tau$	Speedup	Retention (%)
$\lambda$	0.0	3.41	1.40×	100
	0.2	4.12	1.55×	99.3
	0.5	5.83	2.14×	97.8
	0.7	7.27	2.54×	98.5
	0.9	8.97	3.03×	92.8
$N$	1	7.15	2.46×	94.6
	10	7.27	2.54×	98.5
	100	7.47	2.65×	93.4

Table 3: Sensitivity study of LVSPEC (w/o PST) on loose hyperparameter  $\lambda$  and critical visual token  $N$ .

that relaxing a small portion of position shift-induced mismatches is benign.

## 4.3 Sensitivity Study on Hyperparameters

Here, we investigate the two hyperparameters of LVSPEC, as illustrated in Table 3. For the loose parameter  $\lambda$ , we evaluate a range of values in  $[0, 1]$ , spanning verification strategies from conservative to aggressive. For smaller  $\lambda$ , the speedup achieved by LVSPEC increases as  $\lambda$  grows, while accuracy remains  $> 97.8\%$ . Under a highly relaxed setting ( $\lambda = 0.9$ ), excessive relaxation leaves more visual-relevant tokens without strict verification, leading to an accuracy drop to 92.8%. Accordingly, we adopt a moderate choice of  $\lambda = 0.7$ , which strikes a favorable balance between speed and accuracy.

For the number of critical visual tokens  $N$ , we evaluate  $N \in \{1, 5, 10, 100\}$ . Using too few critical tokens (e.g.,  $N = 1$ ) fails to capture sufficient visual cues, whereas using too many (e.g.,  $N = 100$ ) can partially dilute truly salient information (refer to Figure 4), both leading to an accuracy drop. Thus, we adopt a moderate choice of  $N = 10$ .<sup>6</sup>

## 4.4 Which Tokens Are Loosened?

To improve the interpretability of LVSPEC, we present a case study in Figure 8. In the model output, words closely tied to the salient subjects and actions such as “engaging”, “cocktail”, “man”, and “bowls”, are strictly verified, preserving the quality of the video-to-text generation. In contrast, conjunctions, punctuation, and function words such as “depict”, “the”, “:”, and “##” are relaxed, substantially boosting the mean accepted length  $\tau$ . From a broader perspective on vision-to-text tasks, LVSPEC differs from prior work that sparsifies information on the *vision* side of the input. Instead, it opens a new direction by inducing sparsity on the *text* side of the output for orthogonal acceleration.

## 5 Related Work

### 5.1 Speculative Decoding for LLMs

Speculative decoding is shown to be an effective approach to accelerate LLMs while maintaining the

<sup>6</sup>Experiments on draft model sizes and draft tree variants are elaborated in Appendices D and E.

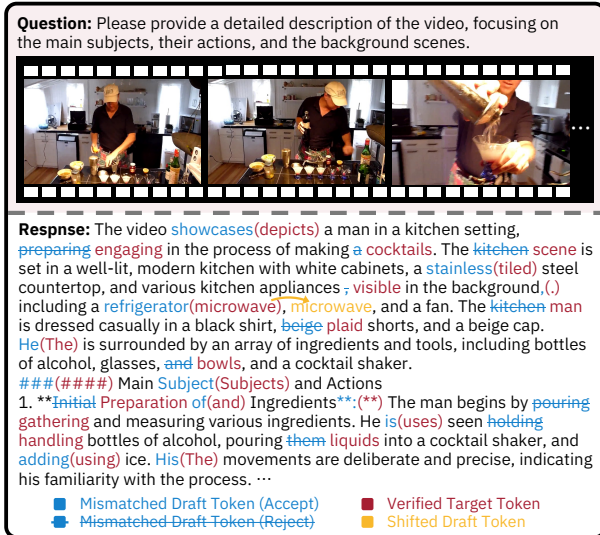


Figure 8: Case study of which tokens are strictly verified and which are relaxed.

original output distribution, following a draft-then-verify paradigm. Initial explorations (Leviathan et al., 2023; Kim et al., 2023; Xia et al., 2023) attempt to use existing small LLMs as draft models to ensure reliable speculation. Self-speculative methods (Xia et al., 2025; Zhang et al., 2024; Song et al., 2025) use partial layers of the original model to generate predictions, without introducing extra models. Retrieval-based methods (He et al., 2024; Luo et al., 2025) retrieve  $n$ -gram continuations as draft tokens. Long-context SD methods (Yang et al., 2025; Chen et al., 2025; Sun et al., 2024) target at long-context scenario. Tree-based speculation methods (Li et al., 2024d,c; Miao et al., 2024) are proposed to boost the average accept length, by predicting multiple candidates and forming draft trees. Recent advancements (Cai et al., 2024; Li et al., 2024d; Du et al., 2024; Zhang et al., 2025b) focus on enhancing the efficiency and accuracy of the draft model through specialized training, and performing parallel drafting and verification (Liu et al., 2025a; Shen et al., 2026b; Kumar et al., 2026).

**Loosely Speculative Decoding** Strict verification often rejects semantically valid drafts, limiting achievable speedups. To address this, recent advances propose “loosely” variants to relax the criterion. JudgeDecoding (Garipov et al., 2025) trains a compact module on top of the embeddings to produce “judgement” of the current continuation. Following studies (Wang et al., 2025b; Zhong et al., 2025; Wang et al., 2025a; Holsman et al., 2025; Byun et al., 2025) further enrich the area of loose verification. Recently, FLY (Li et al., 2025b) proposes a training-free relaxation mechanism based on the entropy of mismatched tokens and whether

the subsequent outputs remain unchanged within a predefined window. These efforts reflect a rising interest in moving beyond strict token-level equivalence, strategically trading exactness for improved efficiency.

## 5.2 Speculative Decoding for LVLMS

Large Vision-Language Models (LVLMS) suffer from slow inference due to their large parameter counts and long input sequences. To address this, prior work (Ji et al., 2025; Kang et al., 2025; Ganesan et al., 2025; Huo et al., 2025; Bajpai and Hanawal, 2025; Xie et al., 2025; Huang et al., 2025; Lin et al., 2025; Wang et al., 2025c; Zhang et al., 2026b; Kong et al., 2026; Shen et al., 2026a) has explored speculative decoding for LVLMS, primarily through vision-aware drafting. For Video-LLMs, SpecVLM (Ji et al., 2025) pioneers in adopting a training-free method, which performs token pruning for the draft model to address the KV cache bottleneck in long video scenario. LVSPEC is the first to perform visual-aware verification, effectively breaking the performance ceiling imposed by the *exact-match* for Video-LLMs.

## 6 Conclusion

We present LVSPEC, the first loosely speculative decoding tailored for Video-LLMs. Thanks to the importance disparity of generated tokens we identified in video understanding, LVSPEC leverages visual-semantic guidance to strictly verify visual-relevant tokens while relaxing criteria for visual-irrelevant ones, effectively overcoming the rigid token-level matching bottleneck. Ultimately, LVSPEC breaks the conventional acceleration ceiling of visual speculative decoding, providing the community with a practical foundation for more efficient and responsive video understanding in real-world applications.

## 7 Acknowledgements

The work was supported by the Major Research Program of the Zhejiang Provincial Natural Science Foundation (Grant No. LD24F020015), CCF-Baidu Open Fund (No. 202509), and Zhejiang Province "Leading Talent of Technological Innovation Program" (No. 2023R5214).

## 8 Limitations

While our loosely speculative decoding framework yields substantial gains in mean accepted length and acceleration for Video-LLMs, several limitations merit consideration. First, since our approach is in a training-free manner, it may require tuning hyper-parameters to better balance output quality and speedup ratio when generalizing to tasks with substantially higher or lower visual information density. Second, our method primarily focuses on the mainstream visually descriptive captioning and QA tasks. For complex visual reasoning tasks that involve more logical tokens, mismatches that are logically important yet visually irrelevant may be potentially handled using a more careful strategy (e.g., further reducing  $\lambda$  or adding auxiliary heuristics). Third, the position shift-tolerant (PST) component is a bonus module tailored to the output patterns of existing video understanding tasks, relying on cases where the draft and target models exhibit positionally shifted outputs. However, since its impact is small, it can be enabled selectively in other settings. Finally, exploring tighter integration with more draft models that are finetuned (Zhang et al., 2025a) or specially trained for LVLMs (Kang et al., 2025; Xie et al., 2025), and extending our framework to more visual understanding tasks is also left for future work.

## 9 Ethical Considerations

All experiments in this work are conducted using open-source datasets and models<sup>7 8 9 10</sup>. Our research focuses solely on improving inference efficiency and does not involve any sensitive data, human subjects, or commercial use.

## References

- Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, and 3 others. 2025. [LLaVA-OneVision-1.5: Fully Open Framework for Democratized Multimodal Training](#). *CoRR*, abs/2509.23661.
- Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Arsiom<sup>7</sup><https://huggingface.co/datasets/wchai/Video-Detailed-Caption> (Apache License 2.0)
- <sup>8</sup><https://huggingface.co/datasets/Enxin/MovieChat-1K-test> (BSD 3-Clause License)
- <sup>9</sup><https://huggingface.co/datasets/lmms-lab/Video-MME> (Creative Commons Attribution-ShareAlike 4.0 International License)
- <sup>10</sup><https://huggingface.co/datasets/lmms-lab/VideoDetailCaption> (CC-BY-4.0 License)
- Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali K. Thabet, and Jonas Kohler. 2025. [Judge Decoding: Faster Speculative Sampling Requires Going Beyond Model Alignment](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. [Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities](#). *CoRR*, abs/2308.12966.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-VL Technical Report](#). *CoRR*, abs/2502.13923.
- Divya Jyoti Bajpai and Manjesh Kumar Hanawal. 2025. [FastVLM: Self-Speculative Decoding for Fast Vision-Language Model Inference](#). *CoRR*, abs/2510.22641.
- Sanghyun Byun, Mohanad Odema, Jung Ick Guack, Baisub Lee, Jacob Song, and Woo Seong Chung. 2025. [3-Model Speculative Decoding](#). *CoRR*, abs/2510.12966.
- Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. [Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.
- Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. 2025. [AuroraCap: Efficient, Performant Video Detailed Captioning and a New Benchmark](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.
- Jian Chen, Vashisth Tiwari, Ranajoy Sadhukhan, Zhuoming Chen, Jinyuan Shi, Ian En-Hsu Yen, and Beidi Chen. 2025. [MagicDec: Breaking the Latency-Throughput Tradeoff for Long Context Generation with Speculative Decoding](#). *The Thirteenth International Conference on Learning Representations*.
- Cunxiao Du, Jing Jiang, Yuanchen Xu, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, and Yang You. 2024. [GLiDe with a CaPE: A Low-Hassle Method to Accelerate Speculative Decoding](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.
- Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong Li Lee, and Wynne Hsu. 2024. [Video-of-thought: step-by-step video reasoning from perception to cognition](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 13109–13125.

- Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. **Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis**. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 24108–24118.
- Mugilan Ganesan, Shane Segal, Ankur Aggarwal, Nish Sinnadurai, Sean Lie, and Vithursan Thangarasa. 2025. **MASSV: Multimodal Adaptation and Self-Data Distillation for Speculative Decoding of Vision-Language Models**. *CoRR*, abs/2505.10526.
- Jun Gao, Qian Qiao, Tianxiang Wu, Zili Wang, Ziqiang Cao, and Wenjie Li. 2025. **Aim: Let any multimodal large language models embrace efficient in-context learning**. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3077–3085.
- Roman Garipov, Fedor Velikonitsev, Ruslan Svirschevski, Vage Egiazarian, and Max Ryabinin. 2025. **AutoJudge: Judge Decoding Without Manual Annotation**. *CoRR*, abs/2504.20039.
- Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. 2025. **Videospresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection**. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26181–26191.
- Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. 2024. **REST: Retrieval-Based Speculative Decoding**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1582–1595.
- Maximilian Holsman, Yukun Huang, and Bhuwan Dhingra. 2025. **Fuzzy Speculative Decoding for a Tunable Accuracy-Runtime Tradeoff**. In *Findings of the Association for Computational Linguistics (ACL)*, pages 26257–26273.
- Haiduo Huang, Fuwei Yang, Zhenhua Liu, Xuanwu Yin, Dong Li, Pengju Ren, and Emad Barsoum. 2025. **SpecVLM: Fast Speculative Decoding in Vision-Language Models**. *CoRR*, abs/2509.11815.
- Mingxiao Huo, Jiayi Zhang, Hwei Wang, Jinfeng Xu, Zheyu Chen, Huilin Tai, and Yijun Chen. 2025. **SpecLLaVA: Accelerating Vision-Language Models with Dynamic Tree-Based Speculative Decoding**. *CoRR*, abs/2509.11961.
- Yicheng Ji, Jun Zhang, Heming Xia, Jinpeng Chen, Lidan Shou, Gang Chen, and Huan Li. 2025. **SpecVLM: Enhancing Speculative Decoding of Video LLMs via Verifier-Guided Token Pruning**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7216–7230.
- Jialiang Kang, Han Shu, Wenshuo Li, Yingjie Zhai, and Xinghao Chen. 2025. **ViSpec: Accelerating Vision-Language Models with Vision-Aware Speculative Decoding**. *CoRR*, abs/2509.15235.
- Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. 2023. **Speculative Decoding with Big Little Decoder**. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*.
- Quan Kong, Yuhao Shen, Yicheng Ji, Huan Li, and Cong Wang. 2026. **ParallelVLM: Lossless Video-LLM Acceleration with Visual Alignment Aware Parallel Speculative Decoding**. *arXiv preprint arXiv:2603.19610*.
- Tanishq Kumar, Tri Dao, and Avner May. 2026. **Speculative Speculative Decoding**. *arXiv preprint arXiv:2603.03251*.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. **Fast Inference from Transformers via Speculative Decoding**. In *International Conference on Machine Learning, ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.
- Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. 2024a. **Lmms-eval: Accelerating the development of large multimodal models**.
- Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2025a. **LLaVA-OneVision: Easy Visual Task Transfer**. *Trans. Mach. Learn. Res.*, 2025.
- Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024b. **LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models**. *CoRR*, abs/2407.07895.
- Jinze Li, Yixing Xu, Guanchen Li, Shuo Yang, Jinfeng Xu, Xuanwu Yin, Dong Li, Edith C. H. Ngai, and Emad Barsoum. 2025b. **Training-Free Loosely Speculative Decoding: Accepting Semantically Correct Drafts Beyond Exact Match**. *Preprint*, arXiv:2511.22972.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024c. **EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7421–7432.

- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024d. **EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty**. In *Forty-first International Conference on Machine Learning*.
- Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. **Video-LLaVA: Learning United Visual Representation by Alignment Before Projection**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984.
- Luxi Lin, Zhihang Lin, Zhanpeng Zeng, and Rongrong Ji. 2025. **Speculative Decoding Reimagined for Multimodal Large Language Models**. *CoRR*, abs/2505.14260.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. **Improved Baselines with Visual Instruction Tuning**. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26286–26296.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. **Visual Instruction Tuning**. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*.
- Tianyu Liu, Yun Li, Qitan Lv, Kai Liu, Jianchen Zhu, Winston Hu, and Xiao Sun. 2025a. **PEARL: Parallel Speculative Decoding with Adaptive Draft Length**. In *The Thirteenth International Conference on Learning Representations*.
- Xuyang Liu, Xiyan Gui, Yuchao Zhang, and Linfeng Zhang. 2025b. **Mixing Importance with Diversity: Joint Optimization for KV Cache Compression in Large Vision-Language Models**. *arXiv preprint arXiv:2510.20707*.
- Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. 2025c. **Video compression commander: Plug-and-play inference acceleration for video large language models**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1910–1924.
- LMMs-Lab. 2024. **Video Detail Caption**. Accessed: 2024-11.
- Xianzhen Luo, Yixuan Wang, Qingfu Zhu, Zhiming Zhang, Xuanyu Zhang, Qing Yang, and Dongliang Xu. 2025. **Turning Trash into Treasure: Accelerating Inference of Large Language Models with Token Recycling**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6816–6831.
- Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2024. **SpecInfer: Accelerating Large Language Model Serving with Tree-based Speculative Inference and Verification**. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.
- Hui Shen, Xin Wang, Ping Zhang, Yunta Hsieh, Qi Han, Zhongwei Wan, Ziheng Zhang, Jingxuan Zhang, Jing Xiong, Ziyuan Liu, Yifan Zhang, Hangrui Cao, Chenyang Zhao, and Mi Zhang. 2026a. **MMSpec: Benchmarking Speculative Decoding for Vision-Language Models**. *arXiv preprint arXiv:2603.14989*.
- Yuhao Shen, Tianyu Liu, Junyi Shen, Jinyang Wu, Quan Kong, Li Huan, and Cong Wang. 2026b. **Double: Breaking the Acceleration Limit via Double Retrieval Speculative Parallelism**. *Preprint*, arXiv:2601.05524.
- Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. 2024. **MovieChat: From Dense Token to Sparse Memory for Long Video Understanding**. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.
- Mingbo Song, Heming Xia, Jun Zhang, Chak Tou Leong, Qiancheng Xu, Wenjie Li, and Sujian Li. 2025. **KNN-SSD: Enabling Dynamic Self-Speculative Decoding via Nearest Neighbor Layer Set Optimization**. *CoRR*, abs/2505.16162.
- Hanshi Sun, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen. 2024. **Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding**. In *First Conference on Language Modeling*.
- Jikai Wang, Zhenxu Tian, Juntao Li, Qingrong Xia, Xinyu Duan, Zhe-Feng Wang, Baoxing Huai, and Min Zhang. 2025a. **Alignment-Augmented Speculative Decoding with Alignment Sampling and Conditional Verification**. *CoRR*, abs/2505.13204.
- Yixuan Wang, Yijun Liu, Shiyu Ji, Yuzhuang Xu, Yang Xu, Qingfu Zhu, and Wanxiang Che. 2025b. **Think Before You Accept: Semantic Reflective Verification for Faster Speculative Decoding**. *CoRR*, abs/2505.18629.
- Zihua Wang, Ruibo Li, Haozhe Du, Joey Tianyi Zhou, Yu Zhang, and Xu Yang. 2025c. **FLASH: Latent-Aware Semi-Autoregressive Speculative Decoding for Multimodal Tasks**. *CoRR*, abs/2505.12728.
- Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023. **Speculative Decoding: Exploiting Speculative Execution for Accelerating Seq2seq Generation**. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 3909–3925. Association for Computational Linguistics.
- Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. 2025. **SWIFT: On-the-Fly Self-Speculative Decoding for LLM Inference Acceleration**. In *The Thirteenth International Conference on Learning Representations*.

- Zhinan Xie, Peisong Wang, and Jian Cheng. 2025. [HiViS: Hiding Visual Tokens from the Drafter for Speculative Decoding in Vision-Language Models](#). *CoRR*, abs/2509.23928.
- Penghui Yang, Cunxiao Du, Fengzhuo Zhang, Haonan Wang, Tianyu Pang, Chao Du, and Bo An. 2025. [LongSpec: Long-Context Speculative Decoding with Efficient Drafting and Verification](#). *CoRR*, abs/2502.17421.
- Bowen Zeng, Feiyang Ren, Jun Zhang, Xiaoling Gu, Ke Chen, Lidan Shou, and Huan Li. 2026. [Hybridkv: Hybrid kv cache compression for efficient multimodal large language model inference](#). *Preprint*, arXiv:2604.05887.
- Jun Zhang, Yicheng Ji, Feiyang Ren, Yihang Li, Bowen Zeng, Zonghao Chen, Ke Chen, Lidan Shou, Gang Chen, and Huan Li. 2026a. [Efficient inference for large vision-language models: Bottlenecks, techniques, and prospects](#). *Preprint*, arXiv:2604.05546.
- Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024. [Draft & Verify: Lossless Large Language Model Acceleration via Self-Speculative Decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11263–11282.
- Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Yang You, Guiming Xie, Xuejian Gong, and Kunlong Zhou. 2025a. [Train Small, Infer Large: Memory-Efficient LoRA Training for Large Language Models](#). *The Thirteenth International Conference on Learning Representations*.
- Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. 2025b. [Learning Harmonized Representations for Speculative Sampling](#). In *The Thirteenth International Conference on Learning Representations*.
- Lei Zhang, Xin Zhou, Chaoyue He, Di Wang, Yi Wu, Hong Xu, Wei Liu, and Chunyan Miao. 2025c. [Mmesgbench: Pioneering multimodal understanding and complex reasoning benchmark for esg tasks](#). In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12829–12836.
- Libo Zhang, Zhaoning Zhang, Wangyang Hong, Peng Qiao, and Dongsheng Li. 2026b. [Sparrow: Text-Anchored Window Attention with Visual-Semantic Glimpsing for Speculative Decoding in Video LLMs](#). *arXiv preprint arXiv:2602.15318*.
- Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025d. [SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference](#). In *Forty-second International Conference on Machine Learning*.
- Meiyu Zhong, Noel Teku, and Ravi Tandon. 2025. [Speeding up Speculative Decoding via Sequential Approximate Verification](#). *CoRR*, abs/2502.04557.

## A Theoretical Derivations for Section 2.2

### A.1 Geometric Acceptance Assumption

**Assumption** (Geometric Acceptance Dynamics). The generation process is inherently autoregressive, where the alignment probability is conditioned on the prefix  $P(\hat{y}_k = y_k \mid y_{<k})$ . Modeling these step-wise dependencies renders the derivation intractable. Thus, for **theoretical tractability**, we simplify the analysis by approximating token correctness as independent and identically distributed (i.i.d.) events with a base alignment accuracy  $\alpha := P(\hat{y}_k = y_k)$ . Consequently, the acceptance process follows a geometric distribution governed by the failure rate  $\epsilon := 1 - \alpha$ .

### A.2 Proof of Proposition 2.1

*Proof.* We analyze the expected value of the accepted length  $\tau_{\text{strict}}^\nabla$ , treating it as a discrete random variable defined over the domain  $\{0, 1, \dots, K\}$ . Invoking the tail sum formula for the expectation of non-negative integer variables, we express  $\mathbb{E}[\tau_{\text{strict}}^\nabla]$  as the sum of cumulative probabilities:

$$\mathbb{E}[\tau_{\text{strict}}^\nabla] = \sum_{k=1}^K P(\tau_{\text{strict}}^\nabla \geq k). \quad (12)$$

The event  $\tau_{\text{strict}}^\nabla \geq k$  is logically equivalent to the successful verification of the first  $k$  consecutive tokens (i.e., the prefix of length  $k$  matches). Under the i.i.d. assumption with alignment accuracy  $\alpha$ , this probability factorizes as:

$$P(\tau_{\text{strict}}^\nabla \geq k) = \prod_{j=1}^k P(\hat{y}_j = y_j) = \alpha^k. \quad (13)$$

Substituting Equation (13) into Equation (12) reveals a finite geometric series characterized by the first term  $\alpha$  and the common ratio  $\alpha$ . Applying the standard closed-form summation formula, we derive the exact expectation:

$$\mathbb{E}[\tau_{\text{strict}}^\nabla] = \sum_{k=1}^K \alpha^k = \frac{\alpha(1 - \alpha^K)}{1 - \alpha}. \quad (14)$$

Crucially, since  $\alpha \in (0, 1)$ , this is a **series of strictly positive terms**. This implies that the partial sum is **strictly monotonically increasing** with respect to  $K$ . Consequently, for any finite draft size  $K$ , the expectation is strictly bounded by the infinite series sum (i.e., the partial sum is strictly less than the infinite sum):

$$\mathbb{E}[\tau_{\text{strict}}^\nabla] < \sum_{k=1}^{\infty} \alpha^k = \frac{\alpha}{1 - \alpha}. \quad (15)$$

We now relate this bound to the failure rate  $\epsilon = 1 - \alpha$ . By substituting  $\alpha = 1 - \epsilon$ , we derive the limit value:

$$\frac{\alpha}{1 - \alpha} = \frac{1 - \epsilon}{\epsilon} = \frac{1}{\epsilon} - 1. \quad (16)$$

Observing that  $\frac{1}{\epsilon} - 1$  is strictly less than  $\frac{1}{\epsilon}$  (since  $\epsilon > 0$ ), we conclude with the strict upper bound:

$$\mathbb{E}[\tau_{\text{strict}}^\nabla] < \frac{1}{\epsilon}. \quad (17)$$

This confirms that the performance of strictly SD is theoretically bottlenecked by the inverse of the draft model’s failure rate.  $\square$

### A.3 Proof of Theorem 2.2

*Proof.* Under the proposed mechanism, tokens in  $\mathcal{A}$  are accepted with probability  $\approx 1$  (conditioned on syntactic plausibility), while tokens in  $\mathcal{V}$  retain the strict acceptance probability  $\alpha$ . This transforms the uniform alignment probability  $\alpha$  into an **effective alignment rate**  $\tilde{\alpha}$ , defined as the weighted average over the sequence:

$$\tilde{\alpha} := \underbrace{\rho \cdot \alpha}_{\nabla_{\text{strict}} \text{ for } \mathcal{V}} + \underbrace{(1 - \rho) \cdot 1}_{\nabla_{\text{loose}} \text{ for } \mathcal{V}'} = 1 - \rho(1 - \alpha). \quad (18)$$

Analogous to the derivation in Proposition 2.1, the expected accepted length follows the geometric expectation formula. By expanding the algebraic term, we establish a strict upper bound:

$$\begin{aligned} \mathbb{E}[\tau_{\text{loose}}^\nabla] &= \sum_{k=1}^K \tilde{\alpha}^k \approx \frac{\tilde{\alpha}}{1 - \tilde{\alpha}} \\ &= \frac{1 - \rho(1 - \alpha)}{1 - (1 - \rho(1 - \alpha))} \\ &= \frac{1 - \rho\epsilon}{\rho\epsilon} = \frac{1}{\rho\epsilon} - 1 < \frac{1}{\rho\epsilon}. \end{aligned} \quad (19)$$

This approximation holds because the visual density is sparse ( $\rho \ll 1$ ), implying that the term  $\frac{1}{\rho\epsilon}$  dominates the subtractive constant. Comparing this to the strict expectation limit  $\mathbb{E}[\tau_{\text{strict}}^\nabla] \approx \frac{1}{\epsilon}$ , we derive the scaling ratio:

$$\frac{\mathbb{E}[\tau_{\text{loose}}^\nabla]}{\mathbb{E}[\tau_{\text{strict}}^\nabla]} \approx \frac{1/\rho\epsilon}{1/\epsilon} = \frac{1}{\rho}. \quad (20)$$

This concludes the proof.  $\square$

## B Benchmark Details

For comprehensive evaluation on video understanding, we conduct experiments on both video captioning and video question answering tasks, where

Benchmark	Type	Output	Frames	#Instances
VDC (Chai et al., 2025)	Captioning	Long	64	120
VDD (LMMs-Lab, 2024)	Captioning	Long	128	30
MovieChat (Song et al., 2024)	QA	Medium	128	100
Video-MME (Fu et al., 2025)	QA	Short	64	500

Table 4: Benchmark details.

the output length ranges from a few tokens to long captions. All evaluations follow the standard implementation of LMMs-Eval (Li et al., 2024a) and use Large Language Models as judges. Prompt template aligns with step 3 in Figure 11. The video captioning benchmarks include ① Video Detail Caption (VDC) (Chai et al., 2025), which covers main objects, backgrounds, details, and camera tests, as well as ② Video Detail Description (VDD) (LMMs-Lab, 2024), which evaluates holistic video-level descriptions. The video question answering benchmarks include ③ MovieChat (Song et al., 2024), which evaluates open-ended question answering over videos, and ④ Video-MME (Fu et al., 2025), which assesses multiple-choice question answering. These benchmarks span diverse task types, video lengths, and output lengths, enabling a comprehensive evaluation of both acceleration and performance on video understanding. We summarize the generation length, number of frames sampled, and number of instances of each benchmark in Table 4.

## C Baseline Details

For our re-implementation of FLY (Li et al., 2025b), we set the size of the deferred window  $Window = 4$ , matching the ratio of  $Window$  to draft length used in the original paper. According to model-specific properties, we set the gate threshold of entropy  $Entropy = 0.1$  for Qwen2.5-VL-32B and  $Entropy = 0.2$  for the other models. The threshold is chosen to induce a relaxation level comparable to that of LVSpec. In Section 4.2, we evaluate more hyperparameter configurations of FLY and FLY<sup>⊖</sup>, and LVSpec consistently remains superior in both speedup and performance (see Figure 7).

For the implementation of SPECVLM (Ji et al., 2025), we follow the original paper and adopt the same tree structure and pruning ratio = 0.9 for visual tokens. The visual tokens are uniformly pruned, and the same attention implementation (scaled-dot-product-attention) is ensured.

## D Exploration on Draft Model Sizes

To assess the generalization of LVSpec under different draft model capacities, we experiment with both larger draft model Qwen2.5-VL-32B, medium

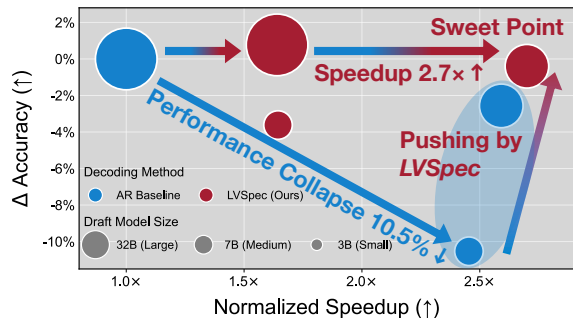


Figure 9: Evaluation of inference efficiency and generation quality. We plot the  $\Delta$  Accuracy against the speedup ratio on the VDC task. Bubble sizes correspond to the draft model scales (3B, 7B, and 32B) within the Qwen2.5-VL family. The proposed LVSpec (red bubbles) demonstrates robust performance scaling, effectively mitigating the accuracy loss observed in the autoregressive decoding (AR) baseline (blue bubbles) even when using weaker draft models for SD.

draft model Qwen2.5-VL-7B, and smaller draft model Qwen2.5-VL-3B, fixing Qwen2.5-VL-32B as the target model on the VDC task, as shown in Figure 9. For draft model Qwen2.5-VL-32B and Qwen2.5-VL-7B, we keep  $\lambda = 0.7$ , and LVSpec still maintains nearly lossless accuracy value (+0.77% and -0.11%) while achieving a speedup ratio of 1.64 $\times$  and 2.70 $\times$ . For draft model Qwen2.5-VL-3B, we adopt a smaller loose parameter  $\lambda = 0.3$  and  $N = 5$  since the draft quality degrades substantially. Even so, the results demonstrate that LVSpec can adaptively preserve accuracy across different model pairings, while delivering meaningful acceleration.

## E Exploration on Draft Token Trees

LVSpec operates on the verification mechanism and is compatible with the draft tree technique commonly used in speculative decoding (Li et al., 2024d,c; Miao et al., 2024). We adopt a simple static tree with two branches (each with draft tokens  $K = 10$ ) and perform loose verification along each branch, finally selecting the branch with the longest accepted length. Table 5 shows that incorporating the tree structure further improves the speedup from 2.70 $\times$  and 2.94 $\times$  to 2.97 $\times$  and 3.30 $\times$ . How-

Metric	Std. -SD-Qwen2.5-VL		Std. -SD-LLaVA-OV	
	w/o Draft Tree	w/ Draft Tree	w/o Draft Tree	w/ Draft Tree
Speedup	2.70×	2.97×	2.94×	3.30×
Retention	99.6%	95.5%	99.8%	96.5%
$\tau$	7.76	8.94	7.34	8.46

Table 5: Experiment of LVSPEC with draft token trees on the VDC benchmark.

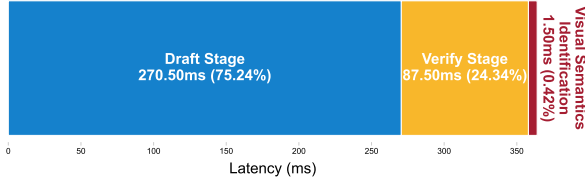


Figure 10: The computational overhead of LVSPEC. Latency is tested using Std. -SD-Qwen2.5-VL and VDC task, on two NVIDIA H200 GPUs.

ever, it in turn increases the level of relaxation, leading to a modest loss in accuracy. We leave the question of how to balance acceptance and speedup across multiple tree branches as future work.

## F Computation Breakdown

In each decoding step, LVSPEC requires computing the visual relevance of text tokens. We report the overhead of this computation in Figure 10. Compared with the cost of drafting and verification, the visual relevance computation accounts for only 0.42% of the runtime, indicating minimal impact on end-to-end latency. Since the cost of computing cosine similarity scales only linearly with the sequence length, this overhead does not become larger relative to the computation of other components as the video length increases.

## G More Experimental Results

**Performance retention results.** In Figure 6, we show the retention results of the performance metric (accuracy and score) for different tasks and model settings. Table 7 provides the raw data.

**More comparison results with SPECVLM.** In the main experiment (Table 1), we compare LVSPEC with SPECVLM (Ji et al., 2025) equipped with the same draft tree as in the original paper. Since the prior loosely SD method (Li et al., 2025b) aims to increase the mean accepted length, it by default use a long chain-structured draft sequence without forming a draft tree. We therefore conduct experiments that compare LVSPEC against SPECVLM under the same draft sequence ( $K = 10$ ) in Table 8. Here, LVSPEC and SPECVLM ( $K = 10$ ) differ only in the verification criterion. The comparison shows that: (i) LVSPEC

surpasses SPECVLM with the same draft structure in both mean accepted length (+128%) and speedup (+93%) solely by loosening the verification. (ii) SPECVLM’s acceleration relies heavily on the draft tree structure, whereas LVSPEC breaks this dependency and is able to accept more tokens with fewer draft nodes, showing potential for larger batch sizes. (iii) The empirical results on mean accepted length closely align with our theoretical analysis in Section 2.2. Specifically, LVSPEC uses  $\lambda = 0.7$  to approximate the setting with visual density  $\rho = 0.3$  in Theorem 2.2. When SPECVLM ( $K = 10$ ) attains  $\tau = 3.41$ , corresponding to a failure rate of 65.9%, LVSPEC achieves a failure rate of  $1 - 7.76/10 = 22.4\%$ , which closely matches the diluted estimate  $65.9\% \times 0.3 = 19.8\%$ .

## Exploration on Tokenizers and Subword Tokens

LVSPEC computes relevance using hidden state representations in the LLM’s internal semantic space, making this process independent of the choice of subsequent tokenizer. Since video LLMs fuse textual and visual information within the LLM backbone, the last-layer hidden states of text and visual tokens can effectively reflect their semantic relevance. In contrast, we also experimented with computing similarity using text and visual embeddings (i.e., the outputs direct from the text tokenizer and the vision encoder, respectively), as in table 6. This strategy performs worse because these embeddings have not undergone semantic alignment within the LLM backbone and thus reside in different representation spaces. By leveraging hidden states as relevance signal, LVSPEC achieves better performance and is independent of a text tokenizer. We believe this intra-model and tokenizer-agnostic design is what enable LVSPEC to generalize across various model settings.

In addition, we also have an interesting finding that the first subword token normally exhibits higher visual relevance, whereas the remaining subwords tend to receive lower relevance scores. We consider this behavior reasonable: earlier subwords often constrain or determine the subsequent ones and therefore carry more semantic information. We do not expect this to cause visually relevant multi-subword words to be misclassified as visually irrelevant, since (1) the first subword token remains

highly discriminative, and (2) later subword tokens are generally unlikely to be mismatched in practice once the first subword is determined.

## H Details of Oracle Study

As mentioned in Section 2.1, we observe that generation quality is governed by a sparse set of “anchor” tokens, whereas high-frequency irrelevant tokens carry minimal information density. A detailed prompt template is provided in Figure 11 and an illustration example is provided in Figure 12.

## I Potential Improvements to LVSPEC

Although the loose verification strategy of LVSPEC demonstrates effective performance across different model settings and benchmarks, it is still necessary to discuss the potential improvements of LVSPEC. We acknowledge that more complex visual reasoning tasks may involve tokens that are visually irrelevant yet logically critical (e.g., “not”), and relaxing the verification of those tokens potentially decreases performance. We discuss this case from the following perspectives: (i) Our method primarily aims to explore the performance boundary of loosely speculative decoding on mainstream video understanding tasks, and to provide an empirically validated solution that is effective under common model settings. Although such cases can arise in theory, the proportion of visually irrelevant but harmful draft tokens is small in descriptive output patterns, and the impact on overall performance is limited according to empirical results (Figure 6 and Table 7). (ii) Potential failure cases are a shared issue in loosely speculative decoding. JudgeDecoding (Bachmann et al., 2025) can degrade in quality when encountering out-of-domain inputs or output patterns unseen during training. FLY (Li et al., 2025b) treats a current mismatch as a synonym if it is followed by several matches, which may inadvertently accept drafts that preserve superficial structural consistency but are nonetheless lower-quality than the target model’s output. (iii) When the output patterns do contain substantial visually irrelevant yet logically critical tokens, improvements can be made upon LVSPEC. First, we can augment the visual relevance metric with additional heuristics to identify non-visual keywords. This would not undermine the role of the visual relevance metric since visually grounded tokens still remain critical. Second, when such tokens cannot be reliably identified, a simple fallback to strict verification can be used to avoid performance degradation.

With the development of complex video reasoning (Fei et al., 2024; Han et al., 2025; Zhang et al., 2025c), incorporating adaptive video token prun-

ing (Liu et al., 2025c; Zhang et al., 2026a), KV cache compression (Liu et al., 2025b; Zeng et al., 2026; Zhang et al., 2026a), and in-context draft capability (Gao et al., 2025) are also promising directions.

## J LLM Usage

Large Language Models (LLMs) were used to aid in the code writing and manuscript polishing. Specifically, the usage includes refining the language, improving readability, and ensuring clarity in the paper. It is important to note that LLMs were not involved in the ideation, research methodology, or experimental design.

Method	Score	Acc (%)	Retention Ratio (%)
LVSPEC (Tokenizer Embedding)	1.88 ± 0.04	29.96 ± 0.79	95.8 ± 2.5
LVSPEC (Hidden State)	1.92 ± 0.02	30.85 ± 0.21	98.7 ± 0.7

Table 6: Exploration on heuristic signals of LVSPEC (w/o PST). The experiment is conducted using Qwen2.5-VL-32B / 7B on the VDC benchmark.  $\lambda = 0.7$ .

Model	Method	VDC		VDD	MovieChat		VideoMME	
		Score	Acc(%)	Score	Score	Acc(%)	Acc <sub>short</sub> (%)	Acc <sub>long</sub> (%)
Std.-SD Qwen2.5-VL	Autoregressive	1.93	31.27	3.98	2.95	54.0	67.6	61.8
	FLY <sup>⊖</sup>	1.87	29.82	3.46	3.11	57.0	67.6	61.8
	FLY	1.94	30.90	3.61	2.82	50.0	67.6	61.8
	LVSPEC (Ours)	1.92	31.16	3.71	3.06	59.0	67.6	61.8
Self-SD Qwen2.5-VL	Autoregressive	1.80	28.70	3.55	3.08	57.0	60.6	59.0
	FLY <sup>⊖</sup>	1.77	27.46	3.70	2.90	51.0	60.6	59.0
	FLY	1.84	29.81	3.63	3.12	58.0	60.6	59.0
	LVSPEC (Ours)	1.82	27.97	3.87	3.04	57.0	60.6	59.0
Std.-SD LLaVA-OV	Autoregressive	1.84	28.66	3.50	2.60	43.0	75.0	65.8
	FLY <sup>⊖</sup>	1.87	29.12	3.53	2.68	48.0	75.0	65.8
	FLY	1.79	27.53	3.03	2.63	43.0	75.0	65.8
	LVSPEC (Ours)	1.83	28.61	3.46	2.53	45.0	75.0	65.8
Self-SD LLaVA-OV	Autoregressive	1.79	27.70	3.57	2.67	48.0	67.0	54.8
	FLY <sup>⊖</sup>	1.84	28.83	3.50	2.58	47.0	67.0	54.8
	FLY	1.87	28.90	3.63	2.61	47.0	67.0	54.8
	LVSPEC (Ours)	1.83	28.38	3.50	2.83	51.0	67.0	54.8

Table 7: Raw data of performance retention results in Figure 6. The overall retention ratio in Figure 6 is computed on average of each performance metric.

Model	Method	Video Detail Caption			Video Detail Description			MovieChat		
		$\tau$	Token/s	Speedup	$\tau$	Token/s	Speedup	$\tau$	Token/s	Speedup
Std.-SD Qwen2.5-VL	SPECVLM ( $K = 10$ )	3.41	11.15	1.40×	3.33	11.85	1.16×	3.02	10.68	1.09×
	SPECVLM	3.29	15.90	2.00×	3.31	17.96	1.76×	3.19	17.05	1.74×
	LVSPEC (Ours)	<b>7.76</b>	<b>21.47</b>	<b>2.70×</b>	<b>7.68</b>	<b>23.58</b>	<b>2.34×</b>	<b>7.74</b>	<b>21.49</b>	<b>2.19×</b>
Self-SD Qwen2.5-VL	SPECVLM ( $K = 10$ )	5.31	25.99	1.26×	5.49	26.03	1.05×	4.63	21.77	0.85×
	SPECVLM	3.85	31.45	1.53×	3.89	35.00	1.41×	3.66	32.20	1.26×
	LVSPEC (Ours)	<b>8.36</b>	<b>36.49</b>	<b>1.77×</b>	<b>8.64</b>	<b>39.32</b>	<b>1.59×</b>	<b>7.99</b>	<b>34.04</b>	<b>1.33×</b>
Std.-SD LLaVA-OV	SPECVLM ( $K = 10$ )	3.66	11.61	1.75×	4.02	10.51	1.75×	4.74	5.49	1.24×
	SPECVLM	3.37	15.75	2.38×	3.53	14.94	2.49×	3.77	8.45	1.92×
	LVSPEC (Ours)	<b>7.34</b>	<b>19.44</b>	<b>2.94×</b>	<b>7.59</b>	<b>17.45</b>	<b>2.91×</b>	<b>8.10</b>	<b>12.16</b>	<b>2.75×</b>
Self-SD LLaVA-OV	SPECVLM ( $K = 10$ )	5.47	26.27	0.99×	5.57	22.61	1.28×	6.98	14.16	0.82×
	SPECVLM	3.91	35.46	1.34×	3.95	29.08	1.65×	4.17	<b>21.19</b>	<b>1.23×</b>
	LVSPEC (Ours)	<b>8.46</b>	<b>36.40</b>	<b>1.38×</b>	<b>8.87</b>	<b>34.96</b>	<b>1.98×</b>	<b>8.77</b>	20.28	1.18×

Table 8: Speedup ratios and mean accepted length  $\tau$ . SPECVLM ( $K = 10$ ) and LVSPEC adopt identical draft structure for fair comparison, while SPECVLM adopts a draft tree as in its original paper for optimization ( $depth = 5$ ). Our LVSPEC outperforms SPECVLM ( $K = 10$ ) by up to 93% and SPECVLM by up to 43% on speedup ratio.

## Oracle Study: Prompt Template (Section 2.1)

**Model Response:** The video frames depict a dynamic and visually rich scene that transitions through various settings, each with distinct elements and activities. Here is a detailed breakdown of the primary objects, their attributes, actions, positions, and movements across the frames:

### **Initial Frames: Modern Office Interior**

1. **Setting:** The scene opens in a modern, well-lit office space with contemporary furniture and decor.

- **Primary Object:** A person is seated on a gray couch, wearing a light-colored shirt and ...



### Step1: Identifying Tokens of Different Visual-Relevance

Please classify the text sequence into three categories and report the frequency of each category.

- Strongly visual-relevant: entities or attributes that are directly observable in the video frames.
- Completely visual-irrelevant: function/grammatical words, abstract terms, and conjunctions.
- Intermediate: tokens that fall between the above two extremes.



### Step2: Remove Visual-Relevant and Visual-Irrelevant Tokens

Please remove the 50 most visually relevant tokens as well as the visually irrelevant tokens from the model output based on the above output and classification results, and return the resulting text after removal.



### Step3: Evaluating the Output Quality

"You are an intelligent chatbot designed for evaluating the correctness of generative outputs for question-answer pairs. "

"Your task is to compare the predicted answer with the correct answer and determine if they match meaningfully. Here's how you can accomplish the task:"

"-----"

"##INSTRUCTIONS: "

"- Focus on the meaningful match between the predicted answer and the correct answer."

"- Consider synonyms or paraphrases as valid matches."

"- Evaluate the correctness of the prediction compared to the answer."

"Please evaluate the following video-based question-answer pair:"

Provide an elaborate description of the main object in the video, covering their attributes, actions, positions, and movements as shown in these frames."

f"Correct Answer:{}"

f"Predicted Answer:{} "

"Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. "

"Please generate the response in the form of a Python dictionary string with keys 'pred' and 'score', where value of 'pred' is a string of 'yes' or 'no' and value of 'score' is in INTEGER, not STRING."

"DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. "

" For example, your response should look like this: {'pred': 'yes', 'score': 4.8}. Please provide the score rounded to two decimal place. "

Figure 11: Prompt template of the oracle study in Section 2.1.

## Oracle Study: Case Analysis (Section 2.1)

**Question:** Describe the primary object or subject in the video, detailing their attributes, actions, positions, and movements in these frames.



Video



**Model Response:** The video frames depict a dynamic and visually rich scene that transitions through various ...

### Step1: Identifying Tokens of Different Visual-Relevance

**Visual-Relevant:** video, scene, settings, frames, modern, office, interior, furniture, décor, person, gray ...

**Visual-Irrelevant:** The, a, that, through, various, each, with, distinct, Here, is, detail, breakdown, of, the ...

### Step2: Remove Visual-Relevant and Visual-Irrelevant Tokens

The video frames depict a dynamic and visually rich scene that transitions through various settings, each with distinct elements and activities. Here is a detailed breakdown of the primary objects, their attributes, actions, positions, and movements across the frames:

### **Initial Frames: Modern-Office Interior\*\***

- Setting\*\*:** The scene opens in a modern, well-lit office space with contemporary furniture and decor.

**Primary Object\*\*:** A person is seated on a gray couch, wearing a light-colored shirt and dark pants.

**Attributes\*\*:** The individual appears relaxed, with their legs crossed and hands resting on their lap.

**Actions\*\*:** The person remains mostly stationary, suggesting a moment of contemplation or rest.

**Position\*\*:** The individual is positioned centrally in the frame, with the couch and surrounding office elements framing the scene.

**Movements\*\*:** Minimal movement, primarily slight shifts in posture or hand placement.
- Background Elements\*\*:**

  - Furniture\*\*:** Modern, sleek design with glass and metal elements.
  - Decor\*\*:** Minimalist, with a few plants and decorative items adding a touch of warmth.
  - Lighting\*\*:** Bright, natural light streaming in from large windows, creating a clean and open atmosphere.

### **Transition to a-Vibrant, Colorful Scene\*\***

- Setting\*\*:** The scene shifts to a lively, colorful environment, possibly a festival or public gathering.

**Primary Object\*\*:** A person is seen in the foreground, wearing a bright, colorful outfit.

**Attributes\*\*:** The individual is engaged in a dynamic activity, possibly dancing or moving energetically.

**Actions\*\*:** The person is actively moving, with arms raised and body swaying.

**Position\*\*:** Central in the frame, drawing attention with their vibrant attire and energetic movements.

**Movements\*\*:** Fluid and rhythmic, suggesting participation in a dance or performance.
- Background Elements\*\*:**

  - Crowd\*\*:** A diverse group of people, also dressed in colorful clothing, contributing to the festive atmosphere.
  - Decor\*\*:** Bright, eye-catching decorations, including flags, banners, and possibly musical instruments.
  - Lighting\*\*:** Bright, possibly enhanced by stage lights, creating a lively and energetic ambiance.

### **Transition to a High-Speed, Blurred Scene\*\***

- Setting\*\*:** The scene changes to a fast-paced, possibly urban environment.

**Primary Object\*\*:** A person is seen in motion, likely riding a bicycle or scooter.

### Step3: Evaluating the Output Quality

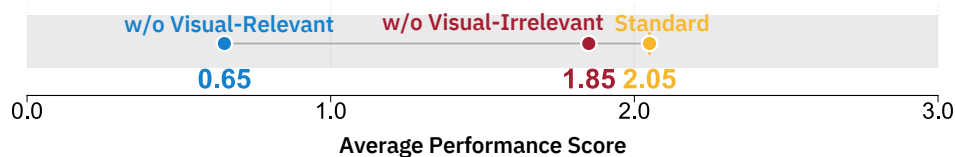


Figure 12: An example of the oracle study in Section 2.1.