SpecVLM: Enhancing Speculative Decoding of Video LLMs via Verifier-Guided Token Pruning

Yicheng Ji[1,2]✌, Jun Zhang[1,2]✌, Heming Xia[3], Jinpeng Chen[4], Lidan Shou[1,2], Gang Chen[1], Huan Li[1,2]✉

[1]The State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3]Department of Computing, The Hong Kong Polytechnic University
[4]School of Computer Science, Beijing University of Posts and Telecommunications

{jiyicheng.cs,zj.cs,should,cg,lihuan.cs}@zju.edu.cn,heming.xia@connect.polyu.hk,jpchen@bupt.edu.cn

Abstract

Video large language models (Vid-LLMs) have shown strong capabilities in understanding video content. However, their reliance on dense video token representations introduces substantial memory and computational overhead in both prefilling and decoding. To mitigate the information loss of recent video token reduction methods and accelerate the decoding stage of Vid-LLMs losslessly, we introduce SpecVLM, a *training-free* speculative decoding (SD) framework tailored for Vid-LLMs that incorporates staged video token pruning. Building on our novel finding that the draft model's speculation exhibits low sensitivity to video token pruning, SpecVLM prunes up to 90% of video tokens to enable efficient speculation without sacrificing accuracy. To achieve this, we perform a two-stage pruning process: Stage I selects highly informative tokens guided by attention signals from the verifier (target model), while Stage II prunes the remaining redundant ones in a spatially uniform manner. Extensive experiments on four video understanding benchmarks demonstrate the effectiveness and robustness of SpecVLM, which achieves up to 2.68× decoding speedup for LLaVA-OneVision-72B and 2.11× speedup for Qwen2.5-VL-32B. Code is available at https://github.com/zju-jiyicheng/SpecVLM.

1 Introduction

Video large language models (Vid-LLMs) (Li et al., 2024a; Ahmed et al., 2025; Lin et al., 2024; Fang et al., 2024) have demonstrated strong performance in video comprehension. Most Vid-LLMs use a sequential visual representation, encoding sampled frames into tens of thousands of video tokens alongside language prompts to achieve high generation performance. However, as long videos become more common, this design introduces significant memory and computational costs. For instance, LLaVA-OneVision (Li et al., 2024a) processes each video frame into 196 tokens, meaning a two-minute video at 60 FPS would require more than 1 million tokens by default without any reduction. The large number of video tokens increases the sequence length, resulting in quadratic attention overhead during prefilling. During decoding, the autoregressive nature of generation exacerbates the memory-bound issue, as the growing key-value (KV) cache must be loaded and stored in GPU memory alongside model parameters, limiting scalability and increasing latency (Lin et al., 2024).

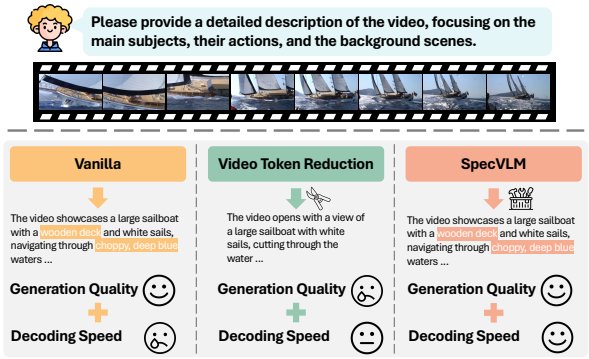

Figure 1: Comparison of vanilla autoregressive decoding, video token reduction, and our SpecVLM.

Recent studies have proposed token pruning strategies (Chen et al., 2024a; Liu et al., 2024; Xing et al., 2025; Zhang et al., 2024b; Tao et al., 2025; Shen et al., 2025; Huang et al., 2025) to mitigate the rapidly growing overhead in storage, access, and computation incurred by tons of *visual* tokens (including *image* tokens and *video* tokens). These methods typically exploit token redundancy and importance variance, applying pruning during the prefilling stage to reduce subsequent memory and compute costs during decoding. However, by physically removing tokens from the input, they incur **inevitable information loss**—especially problem-

✌Equal contribution. ✉Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 7217–7231
November 4-9, 2025

atic in video understanding tasks where rich spatial and temporal cues are essential for maintaining generation quality. In addition, they offer **limited decoding speedup** due to repeated access to full model parameters at each generation step.

Fortunately, speculative decoding (SD) offers a promising solution to accelerate LLM decoding without sacrificing quality (Leviathan et al., 2023) by using a lightweight **draft model** to propose multiple draft tokens, which are then verified in parallel by a **target model**. However, deploying SD for Vid-LLMs is challenging. First, autoregressive draft models for LLMs (Du et al., 2024; Li et al., 2024d,c) suffer from reduced efficiency in video scenarios because they require linearly increasing KV caches, which become the dominant bottleneck as the length of the video grows. Second, the visual context in video is relatively long and sparse, with significant redundancy. While existing SD methodologies tailored for long-context scenarios (Sun et al., 2024; Chen et al., 2025; Yang et al., 2025a) are modality-unaware, they fail to exploit the heavy redundancy and distinct attention patterns of video tokens (detailed in Appendix D), leading to performance degradation (see Section 4.3). These gaps motivate us to perform video token pruning for the draft model, thereby reducing its KV cache size and enhancing speculation efficiency.

Building on the observation that draft model speculation exhibits low sensitivity to random token pruning at low pruning ratios, we propose SpecVLM, a *training-free* speculative decoding framework tailored for Vid-LLMs. As illustrated in Fig. 1, SpecVLM integrates staged video token pruning guided by verifier attention, extending decoding speed gains to high pruning ratios while preserving lossless generation quality. Specifically, we utilize the attention guidance from the target model, and distinguish highly informative video tokens from redundant ones with low attention. Subsequently, we preserve the highly informative tokens following Top-P retention, while discarding the low attention tokens through spatially uniform reduction. By prefilling only the pruned video tokens, the draft model with memory-efficient KV caches achieves enhancing speculation.

To summarize, our main contributions are:

(1) To the best of our knowledge, we are the first to explore speculative decoding for lossless acceleration in Vid-LLMs, and further identify effective video token pruning as the silver bullet for the undesired slowdown in draft model speculation caused by video token explosion.

(2) The surprising insensitivity of draft model speculation to random video token pruning at low pruning ratios sparks the emergence of SpecVLM, a training-free speculative decoding framework with verifier-guided staged video token pruning that pushes the performance boundary under aggressive pruning.

(3) Thorough experiments on four video understanding benchmarks show SpecVLM prunes over 90% of video tokens for the draft model while retaining nearly 90% of the speculation accuracy, achieving up to 2.68× and 2.11× lossless decoding speedup for the LLaVA-OneVision and Qwen2.5-VL, respectively.

2 Preliminary Study

2.1 Naive Speculative Decoding for Vid-LLMs

Given a target model (verifier) $\mathcal{M}_t$ and a draft model $\mathcal{M}_d$, let T_t and T_d be the time for $\mathcal{M}_t$ and $\mathcal{M}_d$ to decode one token. For a predefined speculation length γ, T_t^γ is the time for the target model to verify γ tokens in parallel. Then, the time for each speculation decoding step is written as:

$$T_{step}^{\gamma} = \gamma \cdot T_d + T_t^{\gamma}. \tag{1}$$

Let τ be the average accept length of all decoding steps, the per token time of naive SD is given by:

$$T_{token}^{\gamma} = T_{step}^{\gamma} \,/\, \tau. \tag{2}$$

Hence, the speedup ratio is expressed as:

$$\begin{aligned} \textit{Speedup} &= \frac{T_t}{T_{token}^{\gamma}} = \frac{\tau \cdot T_t}{\gamma \cdot T_d + T_t^{\gamma}} \\ &= \tau \,/\, (\gamma \cdot \frac{T_d}{T_t} + \frac{T_t^{\gamma}}{T_t}). \end{aligned} \tag{3}$$

Eq. (3) reveals that the speedup ratio is affected by (i) average accept length τ, (ii) draft to target latency ratio T_d/T_t, and (iii) verification to target latency ratio T_t^γ/T_t, as described in previous study (Chen et al., 2025). For a normal batch size, T_t^γ/T_t is close to 1 because the GPU parallelism is not fully utilized and the bottleneck still lies in memory bandwidth (Patterson, 2005). Therefore, the speedup ratio primarily depends on (i) and (ii).

Ideally, a draft model should have low latency, leading to a ratio $T_d/T_t \ll 1$. This condition is usually satisfied for LLMs in short-context scenarios by applying a parameter-efficient draft model. However, for Vid-LLMs with long video input, the

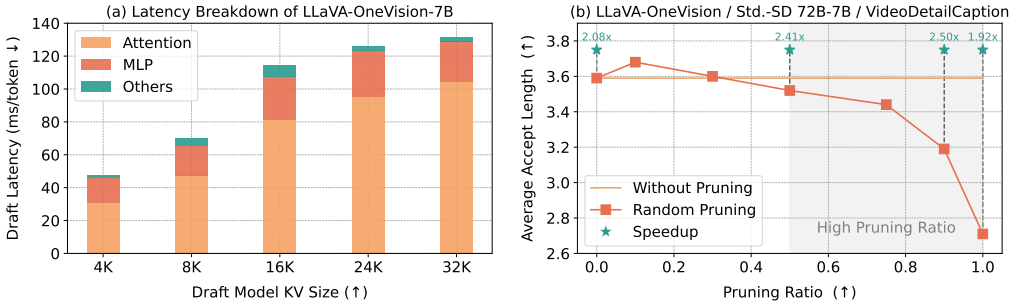

Figure 2: **(a)** Draft latency breakdown of LLaVA-OneVision-7B. Results are measured on a single NVIDIA A100 GPU by averaging the decoding time of 100 tokens. **(b)** Average accept length comparison on standard SD (Std.-SD).

latency bottleneck of the draft model shifts from the parameter scale to the accumulated KV cache, as illustrated in Fig. 2 (a). As the input length grows, the expanding KV cache of the draft model leads to increased draft latency, especially in the attention layers where the entire KV cache is moved from GPU's high-bandwidth memory (HBM) to its on-chip memory (SRAM) in each decoding step (Sun et al., 2024). Hence, reducing the draft model KV size is a straightforward way to cut down draft latency and improve overall speedup for Vid-LLMs.

2.2 Speculation Sensitivity for Token Pruning

To reduce the draft model KV size, we incorporate video token pruning to shorten the video token sequence prefilled by the draft model. However, applying video token pruning may lead to a potential loss of visual information, which would in turn affect the accuracy of speculation. Concretely, as the pruning ratio increases, SD achieves faster speculation from the reduction in draft latency, but at the cost of a decreasing average accept length τ ((i) in Section 2.1). To tackle this trade-off, we aim to address the following key question: *Is the draft model **sensitive** to video token pruning, i.e., can it maintain a stable accept length when part of the visual information is removed?*

To investigate this question, we intuitively apply *random token pruning* for comparison with naive SD policy incorporating tree drafting. We use samples from the VideoDetailCaption (LMMs-Lab, 2024) benchmark and employ LLaVA-OneVision to generate detailed captions of 256 tokens. We report the result in terms of average accept length τ as a direct measurement of speculation accuracy.

Our result in Fig. 2 (b) indicates that random token pruning does not significantly compromise the average accept length under low pruning ratios ($\leq 50\%$), and even leads to improvements at certain pruning ratios. We attribute this to the excessive redundancy in long video input. Numerous redundant tokens divert attention away from important ones, and thus, moderate token removal may yield a positive impact. Moreover, when visual information is entirely removed (100% pruning ratio), we observe a significant drop in the speculation accuracy and overall speedup, indicating that partial retention of video tokens is a better choice. This finding extends the previous work (Gagrani et al., 2024), which suggests that the draft model may not require visual context as input.

To summarize, our experiments demonstrate that **SD is not significantly sensitive to video token pruning under low pruning ratios**, which paves the way for its application. More precisely, it allows us to dramatically diminish the latency ratio T_d/T_t without greatly affecting average accept length τ in Eq. (3), serving as an enhanced solution.

Although random pruning can serve as a decent way to reduce the visual token number (Wen et al., 2025), it suffers from a considerable drop in the average accept length at high pruning ratios (i.e., $> 50\%$), as shown in Fig. 2 (b). We argue that a more efficient and robust method is needed due to the following reasons: (i) As video length grows, methods under lower pruning ratios fail to effectively mitigate the high cost of KV cache. (ii) Random pruning is inherently stochastic and context-agnostic, which provides no deterministic lower bound on performance: with non-zero probability, it may discard all tokens corresponding to a critical semantic element (e.g., object boundaries or scene transitions in information-rich videos). To address these issues, Section 3 introduces a new video token pruning strategy that maintains a high accept length τ across diverse pruning ratios.

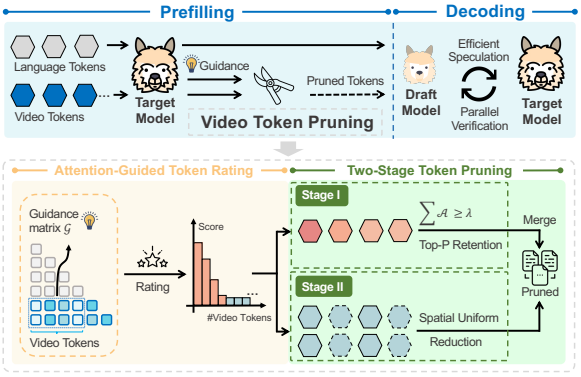

Figure 3: An overview of SPECVLM. In prefilling, the target model guides the pruning of video tokens for the draft model. In decoding, the draft model subsequently makes efficient speculation to accelerate the target model. The pruning process consists of two stages: Top-P retention of highly informative tokens and spatially uniform reduction of low-attention tokens.

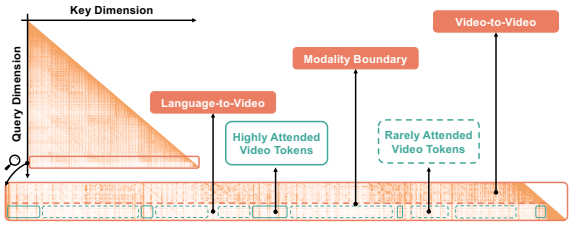

Figure 4: Attention map from LLaVA-OneVision-72B on an input comprising video tokens sampled from four frames in VideoDetailCaption and a language prompt.

3 SPECVLM: Enhancing Speculative Decoding for Vid-LLMs

In this section, we introduce SPECVLM, an enhanced speculative decoding framework incorporated with video token pruning, as depicted in Fig. 3. In Section 3.1, we study the attention pattern of vision-language input, and present our target model's attention-guided token rating scheme. In Section 3.2, we analyze the attention score distribution and propose a two-stage token pruning strategy accordingly. Our method is simple yet effective, and can be applied in a plug-and-play manner.

3.1 Attention-Guided Token Rating from Target Model

To achieve accurate pruning, it is intuitive to preserve the important video tokens related to the query (e.g., the main subjects, actions, and background asked to describe). In previous studies, token importance has been evaluated using criteria such as `[CLS]` score (Shang et al., 2024), attention information from shallow layers (Chen et al., 2024a), or attention maps from small VLMs (Shao et al., 2025; Zhao et al., 2025). We argue that these methods differ from ours in their focus during token pruning: **while they aim to preserve output quality of the original model, our objective is to make the draft model's output better aligned with that of the target model**.

Fig. 3 illustrates how SPECVLM leverages attention signals from the verifier (target model) to guide video token pruning for the draft model. The resulting compact KV cache enables the draft model to perform efficient speculation, accelerating the target Vid-LLM. This design offers key advantages: (i) the target model, being more powerful and structurally similar to the draft, provides more accurate attention signals; (ii) allowing the draft model to "see" where the target model attends helps improve alignment; and (iii) from a latency perspective, the design adds minimal overhead, as target attention is computed regardless of speculation, and the draft only processes the pruned visual input.

To extract accurate attention signals from the target model, we follow previous work (Tu et al., 2025; Li et al., 2025b,a) to study the attention pattern of vision-language input (see Fig. 4). By obtaining the full attention matrix, we observe a clear modality boundary emerging along the query dimension as Tu et al. (2025) described. In particular, video tokens attend to other video tokens densely while language tokens only pay attention to a few video tokens with high concentration, which reflects strong specificity. Moreover, the tokens generated during decoding are language tokens below the modality boundary, where attention patterns exhibit similarity. Based on the above observations, we extract **language-to-video attention scores** from the target model as guidance to rank video tokens for pruning.

Specifically, we take the query-dimensional part of the language modality Q_L together with the key-dimensional part of the vision modality K_V, and the guidance matrix $\mathcal{G}$ is defined by:

$$\mathcal{G} = \mathit{Attention}(Q_L, K_V) = M_{i,j}, \quad \text{and } (i,j) \in \{L, V\}, \tag{4}$$

where L and V denote the language token set and video token set, respectively. Their cardinality are represented by $\|L\|$ and $\|V\|$. Next, we rate video tokens based on the average attention score they received in $\mathcal{G}$. The scores $\mathcal{A}$ are computed to guide

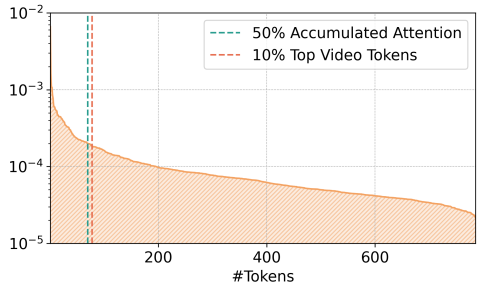

Figure 5: Long-tailed distribution of attention scores.

token pruning, with formulation:

$$\mathcal{A} = \{a_j\}, \text{ where } a_j = \frac{1}{\|L\|} \sum_{i=0}^{i<\|L\|} \mathcal{G}_{i,j}. \quad (5)$$

This operation is performed by averaging across all layers and heads to obtain a holistic assessment.

3.2 Two-Stage Token Pruning for Draft Model

Given an input video token set V and a pruning ratio r, our goal is to prune redundant tokens and retain informative ones for the draft model, guided by the attention scores $\mathcal{A}$. Notably, we observe that the attention scores in $\mathcal{A}$ follow a *long-tailed distribution*, as shown in Fig. 5. This distribution reveals two key insights: (i) A minority of video tokens (10%) accounts for of the total attention large percentage of attention scores (over 50%), which makes their importance stand out compared to other tokens. (ii) The rest of the attention is amortized across the remaining 90% of tokens, making it difficult for the model to differentiate among them, as their attention scores are uniformly low. We attribute this to the high redundancy among video tokens—most carry limited semantic value, yet collectively consume a significant share of attention due to their volume. This observation motivates a binary perspective in our pruning strategy.

Stage I: Top-P Retention of Highly Informative Tokens. To preserve highly influential video tokens, SpecVLM applies Top-P filtering to construct a candidate set whose cumulative attention scores exceed a threshold λ_r ($\lambda_r \in [0, 1]$), determined through a single offline evaluation on a small calibration set with no overlap with the test set [2]. This selection scheme enables dynamic adjustment of the token number for different queries, and ensures that sufficient visual information is retained. Concretely, for a given pruning ratio r, we first sort $\mathcal{A}$, and then repeatedly add tokens to the retention set V_R, until the proportion of their accumulated attention score reaches the predefined threshold λ_r:

$$V_R = \arg\max_{c} \mathcal{A}, \text{ where}$$
$$c = \min \left\{ c' \,\middle|\, \sum_{i=0}^{i<c'} a_i \geq \lambda_r \sum_{i=0}^{i<\|V\|} a_i \right\}. \quad (6)$$

Stage II: Spatially Uniform Reduction of Low Attention Tokens. To handle tokens with uniformly low attention, we exploit the spatial redundancy of video context. As mentioned above, the subtle variations in attention scores make it difficult to distinguish between tokens in the "tail" part in Fig. 5. In Section 4.3, we empirically prove that continuing Top-P retention for these tokens would lead to suboptimal performance. Instead, we leverage the spatial continuity and strong local similarity inherent in video tokens, and select tokens to preserve at a fixed spatial interval I, where:

$$I = \frac{\|V\| - \|V_R\|}{(1-r)\|V\|}. \quad (7)$$

This operation is performed uniformly in space on the remaining token set $V \setminus V_R$. Given that spatially adjacent video tokens are often highly similar, their partial removal incurs minimal visual information loss. Concurrently, retaining them at fixed spatial intervals allows the spatial structure to be effectively preserved. For a comprehensive study, we compare our spatially uniform reduction method with other temporal redundancy-based methods in Section 4.3, and find that spatial redundancy tends to be more significant in our SD setting. We analyze this phenomenon in Section 4.3, and choose to design our approach at the spatial level. Eventually, the tokens chosen in this step are collected as set V_U, which is merged with V_R to form the final retention set of video tokens $V_R \cup V_U$. The draft model is then prefilled using video tokens in $V_R \cup V_U$ along with language prompts, with a KV cache reduced to $1 - r$ of its original size and maximal preservation of video information.

Bonus: Seamless Tree Attention Integration. SpecVLM integrates tree attention by adopting the static tree structure from EAGLE (Li et al., 2024d), implemented via a specialized attention mask[3]. The draft model generates multiple candidate tokens to form a draft tree, which are then verified in parallel by the target model. This design

[2] Model-level selection of λ_r is detailed in Appendix A.

[3] Tree structure is detailed in Appendix B.

Setup	Method	VideoDetailedCaption			MVBench			MVLU			LongVideoBench		
		τ	Tokens/s	Speedup	τ	Tokens/s	Speedup	τ	Tokens/s	Speedup	τ	Tokens/s	Speedup
Std.-SD 72B-7B	Vanilla	-	2.94	-	-	2.93	-	-	3.03	-	-	2.75	-
	SD-Tree	3.68	6.12	2.08×	3.63	5.91	2.02×	3.30	5.75	1.90×	3.57	5.51	2.00×
	SD-Rand	3.19	7.36	2.50×	3.32	6.75	2.30×	2.83	6.03	1.99×	3.14	6.73	2.45×
	SpecVLM	**3.48**	**7.88**	**2.68×**	**3.40**	**6.87**	**2.35×**	**2.97**	**6.06**	**2.00×**	**3.33**	**7.04**	**2.56×**
Self-SD 7B-7B	Vanilla	-	13.31	-	-	12.36	-	-	11.82	-	-	13.55	-
	SD-Tree	4.50	11.25	0.85×	4.52	10.66	0.86×	4.54	10.30	0.87×	4.50	12.10	0.89×
	SD-Rand	3.81	15.73	1.18×	3.93	15.99	1.29×	3.64	13.97	1.18×	3.59	16.83	1.24×
	SpecVLM	**3.98**	**16.74**	**1.26×**	**4.06**	**16.47**	**1.33×**	**3.70**	**14.62**	**1.24×**	**3.84**	**17.63**	**1.30×**

Table 1: Average accepted length τ, decoding speed (tokens/s), and speedup of LLaVA-OneVision series on VideoDetailedCaption, MVBench, MVLU, and LongVideoBench. "Vanilla" refers to vanilla autoregressive decoding while "SD-Tree" denotes speculative decoding with draft trees. "SD-Rand" denotes SD-Tree incorporated with random token pruning. By default, $r = 90\%$.

Setup	Method	VideoDetailedCaption			MVBench			LongVideoBench		
		τ	Tokens/s	Speedup	τ	Tokens/s	Speedup	τ	Tokens/s	Speedup
Std.-SD 32B-7B	Vanilla	-	4.88	-	-	5.50	-	-	4.91	-
	SD-Tree	3.27	6.83	1.40×	3.18	7.56	1.37×	3.24	6.82	1.39×
	SD-Rand	3.21	9.93	2.03×	3.17	10.13	1.84×	3.23	10.20	2.08×
	SpecVLM	**3.23**	**9.99**	**2.05×**	**3.18**	**10.17**	**1.85×**	**3.28**	**10.35**	**2.11×**
Self-SD 7B-7B	Vanilla	-	10.41	-	-	12.28	-	-	9.63	-
	SD-Tree	4.38	8.65	0.83×	4.31	10.02	0.82×	4.17	7.82	0.81×
	SD-Rand	3.75	15.08	1.44×	3.89	16.72	1.36×	3.77	14.49	1.50×
	SpecVLM	**3.83**	**15.59**	**1.50×**	**3.92**	**16.80**	**1.37×**	**3.84**	**15.33**	**1.59×**

Table 2: Average accepted length τ, decoding speed, and speedup of Qwen2.5-VL series on VideoDetailedCaption, MVBench, and LongVideoBench (the sampling protocol of MVLU is incompatible). By default, $r = 90\%$.

improves decoding speed by enabling more tokens to be accepted per forward pass.

4 Experiments

Target and Draft Models. We select two widely used Vid-LLM series: LLaVA-OneVision (Li et al., 2024a) and Qwen2.5-VL (Ahmed et al., 2025). Two representative SD settings are employed. (i) **Standard Speculative Decoding (Std.-SD)**: using a smaller Vid-LLM from the same model series as the draft model. (ii) **Self-Speculative Decoding (Self-SD)**: using the model with original parameters and pruned KV cache as draft model. This setting avoids introducing additional models, allowing the acceleration gains to come solely from video token pruning, which can serve as a more general solution. We do not introduce a training process for the draft model as the major bottleneck is KV caches rather than model parameters, which aligns with Chen et al. (2025); Yang et al. (2025a).

Tasks and Benchmarks. Since SpecVLM mainly focuses on the acceleration during decoding, we evaluate its performance in video captioning and video description tasks, which require generating long text paragraphs. The benchmarks we use include VideoDetailCaption (LMMs-Lab, 2024), MVBench (Li et al., 2024b), MVLU (Zhou et al., 2025a), and LongVideoBench (Wu et al., 2024). Experimental details are elaborated in Appendix A.

Metrics. We assess the acceleration effects using: (i) decoding speed and wall time speedup ratio relative to vanilla autoregressive decoding, and (ii) average accept length τ, which directly reflects the speculation accuracy. Output quality is not evaluated since our method is lossless.

4.1 Main Result

We evaluate the efficiency of different baselines in Tables 1 and 2. For LLaVA-OneVision series, SpecVLM achieves a speedup ratio up to 2.68× and 1.33× under Std.-Sd and Self-SD, respectively. For Qwen2.5-VL series, up to 2.11× and 1.59× speedup are attained accordingly. The default pruning ratio is set to 90% to maximally reduce the KV cache size. Still, we include experiments on different pruning ratios in Section 4.2. The case study and computation breakdown of SpecVLM are elaborated in Appendices C and F.

Token pruning significantly enhances the performance of speculative decoding. Compared to vanilla autoregressive decoding, SD-Tree yields a basic speedup through the draft-and-verify process. When incorporated with video token pruning, the speedup ratio is largely boosted due to

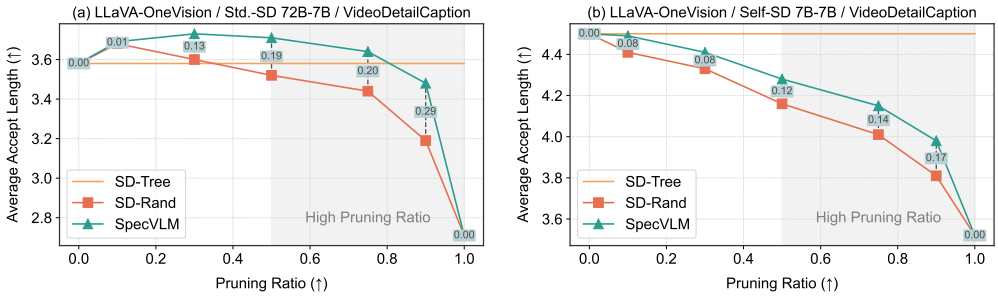

Figure 6: Average accepted length across different baselines as pruning ratios scale up.

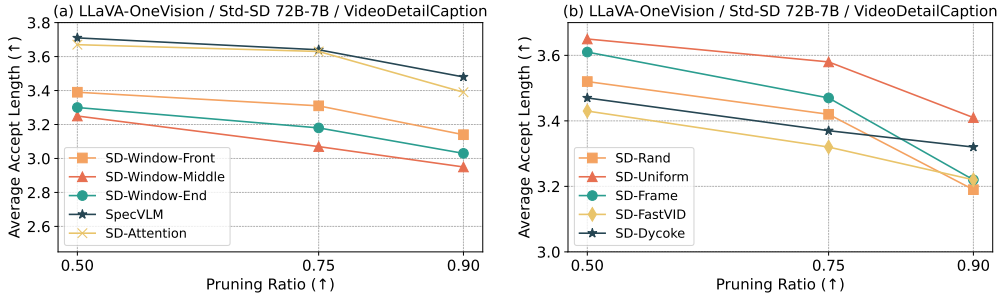

Figure 7: **(a)** SpecVLM vs. window-based methods. **(b)** Spatial vs. temporal redundancy-based methods.

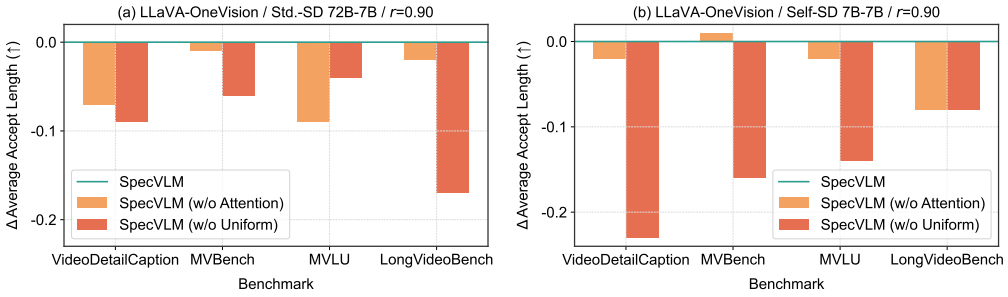

Figure 8: Change in average accepted length (Δ) of the full SpecVLM and its ablated variants.

the utilization of an enhanced draft model with reduced KV cache. For Std.-SD, video token pruning improves decoding speed by 29% and 43% for LLaVA-OneVision-72B and Qwen2.5-VL-32B on VideoDetailCaption compared to SD-Tree. For Self-SD, video token pruning facilitates a 1.24× to 1.33× speedup for LLaVA-OneVision-7B and a 1.37× to 1.59× speedup for Qwen2.5-VL-7B. Additionally, we further note that the influence of video token pruning would become even greater as the video length increases, which can be inferred from the trend revealed in Fig. 2 (a).

SpecVLM shows superior performance across various datasets and model architectures. Tables 1 and 2 illustrate that SpecVLM outperforms random token pruning under a high pruning ratio. On VideoDetailCaption, the average accept length τ of LLaVA-OneVision-7B using our method drops by only 5% under 90% video token reduction, which is 40% of the degradation observed with SD-Rand. For Qwen2.5-VL series and on other datasets, although the sensitivity to pruning is lower, our method consistently demonstrates higher speculation accuracy to varying degrees. Moreover, a higher value of average accept length τ in SpecVLM also implies a higher theoretical upper bound on overall speedup, demonstrating the potential of our method to be integrated with other draft model optimization techniques.

4.2 Scaling Law for Pruning Ratio

For a comprehensive study, we conduct the experiments across a wide spectrum of pruning ratios as shown in Fig. 6, in which SpecVLM achieves a consistently higher speculation accuracy. Moreover, as the pruning ratio increases, the gap between SpecVLM and random token pruning gradually widens, demonstrating the effectiveness of SpecVLM under high pruning ratios.

Decoding Step	0	1	2	3	4	5	6	7	8	9	Avg (First 10 Steps)	Avg (All Steps)
Average Accepted Length	4.3	3.1	3.7	3.4	3.7	2.95	3.6	3.6	2.5	3.1	3.39	3.48

Table 3: Average accepted length vs. decoding steps.

4.3 Ablation Study

Necessity of Attention Guidance. Here, we compare our method against attention-agnostic pruning to validate the necessity of attention guidance. We adopt a window-based approach commonly similar to StreamingLLM cache (Xiao et al., 2024), which retains initial and trailing language tokens along with video tokens within a fixed window. Specifically, we place the window over the front, middle, and end portions of the video tokens, retaining a fixed number of tokens according to the pruning ratio. Fig. 7 (a) shows that our attention-based method (SD-Attention, i.e., SpecVLM without Stage II) outperforms window-based approaches by enabling finer-grained token selection and higher speculation accuracy under a fixed KV budget.

Impact of Top-P Informative Token Retention. To verify the effectiveness of our Top-P informative token retention strategy, we adopt uniform pruning in space without the guidance of the attention signal, as depicted in Fig. 8. In most cases, preserving highly informative tokens improves average accept length. For LLaVA-OneVision / Self-SD 7B-7B, an outlier appears on MVBench, which we attribute to numerous repetitive action sequences in its subset, resulting in more pronounced spatial redundancy.

Impact of Spatially Uniform Token Reduction. Similarly, we remove the second stage of our pruning strategy in Fig. 8 to study the effect of spatially uniform token reduction. The variant of relying solely on attention signals for token retention suffers from indistinguishability issues in the uniformly low-attention area and leads to the loss of structural information and a drop in speculation accuracy, further supporting our observation in Fig. 5.

4.4 Exploration of Diverse Pruning Criteria

Inspired by recent work (Tao et al., 2025; Shen et al., 2025) that explores temporal redundancy, we aim to address the question: *Can pruning based on temporal redundancy offer greater benefits?* To this end, we compare the following baselines:

(1) SD-Uniform: SD with uniform pruning based on token positions within the spatial layout.

(2) SD-Frame: SD with full-frame dropping at regular temporal intervals.

(3) SD-FastVID: Frame-level pruning based on Top-K similarity of consecutive frame transitions, following FastVID (Shen et al., 2025).

(4) SD-DyCoke: Token-level temporal merging adapted from DyCoke (Tao et al., 2025).

Fig. 7 (b) suggests that temporal redundancy-based methods consistently underperform compared to spatially uniform pruning (our focus)—and even random pruning—within the SD framework. We believe this indicates that the draft model benefits from retaining the overall temporal structure and distribution of camera shots to achieve better alignment, while some spatial information is more redundant in this context.

4.5 Impact of Decoding Steps

To validate the effectiveness of our method in selectively preserving visual information to enhance speculation accuracy, we performed experiments on the average acceptance length vs. decoding steps using LLaVA-OneVision-72B/7B with a 90% pruning ratio setting on VideoDetailCaption. We averaged the performance over 50 instances for evaluation. The results are presented in Table 3, from which we can conclude: (i) **The average accept length of the initial steps does not significantly differ from the average across all steps.** The average for the first 10 decoding steps (3.39) is not notably lower than the overall average (3.48), and there is no significant upward trend within the first 10 steps. We attribute this to the fact that each decoding step benefits from the visual information retained during the prefill stage, allowing for a relatively high speculation accuracy from the outset. (ii) **The first decoding step, in fact, exhibits a significantly higher average accepted length.** This is likely because the beginning of the generation often follows a fixed sentence structure, making it easier to speculate.

5 Related Work

5.1 Speculative Decoding

Speculative Decoding for LLMs. Speculative decoding is shown to be an effective approach to

accelerate LLMs while maintaining the original output distribution. It relies on two key processes: efficient drafting and parallel verification. Initial explorations (Leviathan et al., 2023; Guo 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., 2024a; Song et al., 2025) use partial layers of the original model to generate predictions, without introducing extra models. Recent advancements (Cai et al., 2024; Li et al., 2024d; Du et al., 2024) focus on enhancing the efficiency of the drafting stage. These works attain high acceleration ratios through a specially trained draft model with reduced latency. Meanwhile, tree-based speculation methods (Li et al., 2024c,d; Miao et al., 2024) are proposed to boost the average accept length, by predicting multiple candidates and forming draft trees. Among the aforementioned draft models, the success of the previous state-of-the-art EAGLE method (Li et al., 2024c) highlights the *autoregressive structure* as a key factor in improving the accuracy of the speculation. However, autoregressive draft models need to maintain their own KV caches, which introduces additional memory overhead when faced with long video input.

Long-Context Speculative Decoding. Long-context speculative decoding methodologies (Sun et al., 2024; Chen et al., 2025; Yang et al., 2025a) are specially developed to mitigate the substantial overhead of KV cache. Specifically, TriForce (Sun et al., 2024) introduces a hierarchical speculation to tackle the two bottlenecks of model weights and KV cache. MagicDec (Chen et al., 2025) reassesses the trade-off between throughput and latency. LongSpec (Yang et al., 2025a) proposes a memory-efficient draft model with a constant memory footprint. Appendix D elaborates on the reasons why these methods are not suitable for Vid-LLMs.

5.2 Visual Token Reduction

Compared to information-dense text, visual tokens often exhibit high redundancy, making token reduction an effective way to reduce computational and memory overhead. Recent studies largely focus on training-free visual token reduction methods based on token importance and redundancy. FastV (Chen et al., 2024a) selects important visual tokens after layer 2 using attention scores of MLLMs. While SparseVLM (Zhang et al., 2024b) evaluates the visual relevance of text tokens and performs pruning based on the attention scores of a subset of text tokens. VisionZip (Yang et al., 2025b) reduces visual redundancy in the vision encoders. DART (Wen et al., 2025) prunes tokens based on its duplication with other tokens. Currently, video token reduction draws increasing attention due to the high volume of video tokens. DyCoke (Tao et al., 2025) performs token merging across frames and reduces KV cache dynamically. PrunVID (Huang et al., 2025) identifies static and dynamic tokens across frames, and selectively prunes visual features relevant to question tokens. FastVID (Shen et al., 2025) partitions frames into segments and introduces a density-based token pruning strategy. VidCom^2 (Liu et al., 2025) dynamically adjusts compression intensity based on frame uniqueness. Notably, recent Tang et al. (2025) applies temporal processing during video sampling by performing key frame extraction, whereas ours prunes visual tokens post-encoder, making the two methods orthogonal.

6 Conclusion

We propose SpecVLM, the first *training-free* speculative decoding framework tailored for accelerating video LLMs. Building on the low speculation sensitivity to token pruning, SpecVLM leverages verifier-guided attention to remove redundant video tokens, significantly reducing the draft model's KV cache without compromising generation quality. SpecVLM achieves up to 2.68× speedup on LLaVA-OneVision-72B and 2.11× on Qwen2.5-VL-32B across multiple video understanding tasks. We hope our work inspires further research on latency-efficient Vid-LLM reasoning from a token sparsity perspective and believe SpecVLM will serve as a valuable tool for the community to enable efficient video comprehension.

7 Acknowledgements

The work was supported by the Major Research Program of Zhejiang Provincial Natural Science Foundation (No. LD24F020015), Zhejiang Province "Leading Talent of Technological Innovation Program" (No. 2023R5214), Beijing Natural Science Foundation(Grant No.L233034), and Fundamental Research Funds for the Beijing University of Posts and Telecommunications (Grant No.2025TSQY01).

8 Ethical Considerations

All experiments in this work are conducted using open-source datasets and models. Our research focuses solely on improving inference efficiency and does not involve any sensitive data, human subjects, or commercial use [4] [5] [6] [7].

9 Limitation

While our enhanced speculative decoding framework brings clear benefits for accelerating Vid-LLMs, there are a few limitations to consider. First, our method is primarily applicable to resource-constrained long-video scenarios, where memory bandwidth becomes the dominant bottleneck. Second, we introduce an additional draft model during inference. Although its computational overhead is relatively small compared to the target model (or can be avoided using the Self-SD setting), the choice of the draft model requires careful consideration to achieve optimal speedup. Moreover, to avoid introducing training as an additional variable, we use an existing Vid-LLM as the draft model without further fine-tuning. This design choice imposes certain constraints on the maximum achievable acceleration. Nevertheless, our method has the potential to be seamlessly integrated with smaller and faster draft models, as it only requires a one-time pruning of the draft model's KV cache during the prefilling stage. As the technology for training lightweight Vid-LLMs is still in its early stages and lacks suitable candidates, we believe future work can explore more draft model optimization techniques (Li et al., 2024d; Du et al., 2024) to address this limitation.

References

Imtiaz Ahmed, Sadman Islam, Partha Protim Datta, Imran Kabir, Naseef Ur Rahman Chowdhury, and Ahshanul Haque. 2025. Qwen 2.5: A comprehensive review of the leading resource-efficient llm with potentioal to surpass all competitors. *Authorea Preprints*.

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 *International Conference on Machine Learning*, pages 5209–5235. PMLR.

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*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, and 1 others. 2024b. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*.

Cunxiao Du, Jing Jiang, Xu Yuanchen, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, and 1 others. 2024. Glide with a cape: a low-hassle method to accelerate speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11704–11720.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. 2024. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37:89098–89124.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. 2025a. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *Preprint*, arXiv:2407.11550.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. 2025b. Identify critical kv cache in llm inference from an output perturbation perspective. *Preprint*, arXiv:2502.03805.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. 2024. On speculative decoding for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8289.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. 2023. Longcoder: A long-range pre-trained language model for code completion. In *International Conference on Machine Learning*, pages 12098–12107. PMLR.

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

[4] https://huggingface.co/datasets/MLVU/MVLU (CC-BY-NC-SA-4.0 license)

[5] https://huggingface.co/datasets/OpenGVLab/MVBench (CC-BY-4.0 license)

[6] http://github.com/longvideobench/LongVideoBench (CC-BY-NC-SA-4.0 license)

[7] https://huggingface.co/datasets/lmms-lab/VideoDetailCaption (CC-BY-4.0 license)

Xiaohu Huang, Hao Zhou, and Kai Han. 2025. Prunevid: Visual token pruning for efficient video large language models. In *ACL*.

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W Mahoney, Amir Gholami, and Kurt Keutzer. 2023. Speculative decoding with big little decoder. *Advances in Neural Information Processing Systems*, 36:39236–39256.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Mingxiao Li, Na Su, Fang Qu, Zhizhou Zhong, Ziyang Chen, Yuan Li, Zhaopeng Tu, and Xiaolong Li. 2025a. Vista: Enhancing vision-text alignment in mllms via cross-modal mutual information maximization. *Preprint*, arXiv:2505.10917.

Yucheng Li, Huiqiang Jiang, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Amir H Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and 1 others. 2025b. Mapsparse: Accelerating pre-filling for long-context visual language models via modality-aware permutation sparse attention. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

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 *International Conference on Machine Learning*, pages 28935–28948. PMLR.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi 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.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. 2025. Video compression commander: Plug-and-play inference acceleration for video large language models. *arXiv preprint arXiv:2505.14454*.

LMMs-Lab. 2024. Video detail caption. *Accessed: 2024-11*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2024. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *ASPLOS*, pages 932–949.

Professor David Patterson. 2005. Latency lags bandwidth. In *Proceedings of the 2005 International Conference on Computer Design*, pages 3–6.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. 2025. Growing a twig to accelerate large vision-language models. *arXiv preprint arXiv:2503.14075*.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. 2025. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*.

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. *Preprint*, arXiv: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. *COLM*.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. 2025. Adaptive keyframe sampling for long video understanding. In *CVPR*.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. In *CVPR*.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. 2025. VL-cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *The Thirteenth International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857.

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 *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and 1 others. 2025. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CVPR*.

Penghui Yang, Cunxiao Du, Fengzhuo Zhang, Haonan Wang, Tianyu Pang, Chao Du, and Bo An. 2025a. Longspec: Long-context speculative decoding with efficient drafting and verification. *arXiv preprint arXiv:2502.17421*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025b. Visionzip: Longer is better but not necessary in vision language models. *CVPR*.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024a. 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, Gang Chen, Qin Xie, Guiming Xie, and Xuejian Gong. 2025a. HMI: hierarchical knowledge management for efficient multi-tenant inference in pretrained language models. *The VLDB Journal*, 34(4):43.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Yang You, Guiming Xie, Xuejian Gong, and Kunlong Zhou. 2025b. Train Small, Infer Large: Memory-Efficient LoRA Training for Large Language Models. *The Thirteenth International Conference on Learning Representations*.

Jun Zhang, Jue Wang, Huan Li, Zhongle Xie, Ke Chen, and Lidan Shou. 2025c. CHASe: Client Heterogeneity-Aware Data Selection for Effective Federated Active Learning . *IEEE Transactions on Knowledge & Data Engineering*, 37(06):3088–3102.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Zhikai Li, Yibing Song, Kai Wang, Zhangyang Wang, and Yang You. 2025. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. *CVPR*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025a. Mlvu: A comprehensive benchmark for multi-task long video understanding. *CVPR*.

Yuxin Zhou, Zheng Li, Jun Zhang, Jue Wang, Yiping Wang, Zhongle Xie, Ke Chen, and Lidan Shou. 2025b. FloE: On-the-Fly MoE Inference on Memory-constrained GPU. *Forty-second International Conference on Machine Learning*.

A Experimental Details

Task and Benchamark Details. Since SPECVLM mainly focuses on the lossless acceleration during the decoding stage, we select video captioning and video description as our tasks, which require the model to summarize and describe the video by understanding detailed subjects and actions, and involve generating relatively long text paragraphs. Unlike conventional video question answering (VQA), we prompt the model to generate a description rich in visual information instead of direct answer selection. We sample videos from mainstream video understanding benchmarks, including VideoDetailCaption (LMMs-Lab, 2024), MVBench (Li et al., 2024b), MVLU (Zhou et al., 2025a), and LongVideoBench (Wu et al., 2024), to ensure a comprehensive coverage of different durations and varying scenarios. For each benchmark, 50 instances are randomly sampled. For LLaVA-OneVision-72B and LLaVA-OneVision-7B, we uniformly sample 64 and 128 frames to generate a 196×64 and 196×128 video token input, respectively. For Qwen2.5-VL series, we adjust the FPS to generate input of comparable length.

Implementation Details. All experiments are implemented on 8 NVIDIA A100 GPUs. We utilize the default attention implementation of LLaVA-OneVision (scaled-dot-product-attention (Vaswani et al., 2017)), and output the attention scores using its Python implementation when required. Given each query, the target model is employed to generate 256 tokens following greedy decoding. When video tokens are pruned, we remove the corresponding video features based on the pruning ratio r. During evaluation, the default pruning ratio r is set to 90%, based on the low sensitivity property validated in Section 2.2. The hyperparameter λ_r is determined at the model level through a single offline evaluation on a small calibration set consisting of 2–4 randomly sampled instances per task (with no overlap with the test set). In Tables 1 and 2, we set λ_r to 0.4 and 0.5, respectively.

B Impact of Tree Attention

As illustrated in Table 4, using a tree structure for drafting and verification greatly enhances the average accept length by validating multiple candidates in a single forward. When incorporated with tree attention, SPECVLM achieves a higher speculation accuracy. The draft tree structure adopted in SPECVLM is depicted as Fig. 9. The chosen tree structure is motivated by the intuition that candidate tokens with higher probabilities merit deeper and wider expansion, whereas low-probability tokens should not be further explored.

Method	τ
SD-Chain	3.12
SD-Tree	3.68
SPECVLM-Chain	2.79
SPECVLM	3.48

Table 4: Average accept length τ. "*-Chain" refers to a sequential drafting process without a tree structure. Results are tested using LLaVA-OneVision-72B / 7B on VideoDetailCaption. By default, $r = 90\%$.

Operation	Vanilla	SPECVLM
Target Model Prefilling	24.01	24.01
Target Model Decoding	87.03	32.47
Draft Model Prefilling	-	0.82
Video Token Pruning	-	0.06
Latency	111.04	57.36

Table 5: Inference time breakdown (s) of LLaVA-OneVision-72B / 7B. Output length is set to 256.

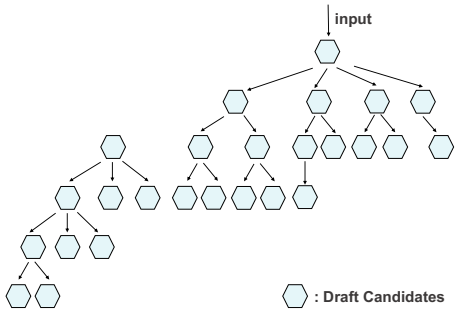

Figure 9: The draft tree structure of SPECVLM.

C Breakdown of Computation

Apart from accelerating the decoding stage of target Vid-LLMs, SPECVLM introduces minimal overhead in the pruning process, as illustrated in Table 5. Owing to video token pruning, the prefill length of the draft model is substantially reduced, and the additional prefilling time becomes negligible compared to target model inference time. In this work, we significantly reduce the target model's decoding time—a level of efficiency that cannot be achieved by prior methods relying solely on token reduction.

D Why is Long-context SD Technologies Not Suitable for Vid-LLMs?

As mentioned earlier in Section 5, long-context SD techniques are designed to address the KV cache bottleneck of conventional SD, combined with KV cache sparsification techniques (Feng et al., 2025a,b; Xiao et al., 2024). To this end, they employ StreamingLLM caches (Xiao et al., 2024) or sliding window caches for the draft model to manage the KV cache budget. However, the lack of awareness of the visual modality prevents these methods from effectively operating on video tokens. On the one hand, video tokens exhibit substantial redundancy, with a large portion being repetitive or highly similar—unlike discrete language tokens. On the other hand, there exists a distinct attention pattern between language and video tokens, as illustrated in Fig. 4, making modality-aware methods better suited to exploit this property. If StreamingLLM-style caches or sliding windows are applied naively, the draft model can only attend to a small portion of the uncompressed visual content, thereby failing to capture the overall semantics of the video.

E Discussion on Smaller Draft Model

In this section, we discuss the effectiveness of SpecVLM in a smaller draft model scenario. Although the minimum draft model used in our experiment is 7b, we believe our method still has benefits for smaller draft models from the following two perspectives. (i) **SpecVLM prunes redundant video tokens at inference time, reducing the training cost and architectural complexity of high-quality small draft models for video understanding.** Existing small draft models for LLMs like EAGLE do not support long-context input sequences. Training small draft models to match the context length of Vid-LLMs would require massive computation, and it remains questionable whether these small models can achieve the same accuracy at video input context lengths (Sun et al., 2024). Therefore, reducing the input length of the small draft model through our video token pruning strategy allows it to perform efficient speculation without the need for costly long-context training. (ii) **For smaller draft models, the overhead caused by input length still persists.** To simulate a scenario similar to EAGLE, we tested the latency of a single layer of the LLaVA-OneVision-0.5B model at varying input lengths. Latency is calculated on average of 100 decoded tokens on a single Nvidia A100 GPU. The results in Table 6 show that the latency of the small draft model increases significantly with input length, eventually becoming multiple times larger than the original. Given the increasing trend of video inputs (e.g., Long video input with million-level video tokens (Chen et al., 2024b)), the additional draft overhead from the KV Cache would gradually undermine the lightweight nature of smaller draft models. In other words, combining draft models with token pruning still provides a performance benefit. In future work, we will investigate memory- and data-efficient training strategies for Vid-LLMs (Li et al., 2024d; Hu et al., 2022; Zhang et al., 2025b,c), with the goal of constructing more optimized draft models that remain effective in memory-constrained devices (Zhou et al., 2025b) and scalable to multi-tenant settings (Zhang et al., 2025a).

Input Sequence Length (K)	1	4	8	16	32	48
Draft Latency (ms)	1.0	1.16	1.46	2.08	3.29	4.53

Table 6: Layer latency variation of smaller draft model with growing input sequence length.

F Case Study

Two illustrative examples are presented in Figs. 10 and 11. The target model performs verification using the original video tokens, ensuring lossless input. Meanwhile, the draft model relies on a 90% pruned version of the video tokens, allowing for efficient yet accurate speculation by retaining essential visual cues and maintaining a coherent understanding of the overall video structure.

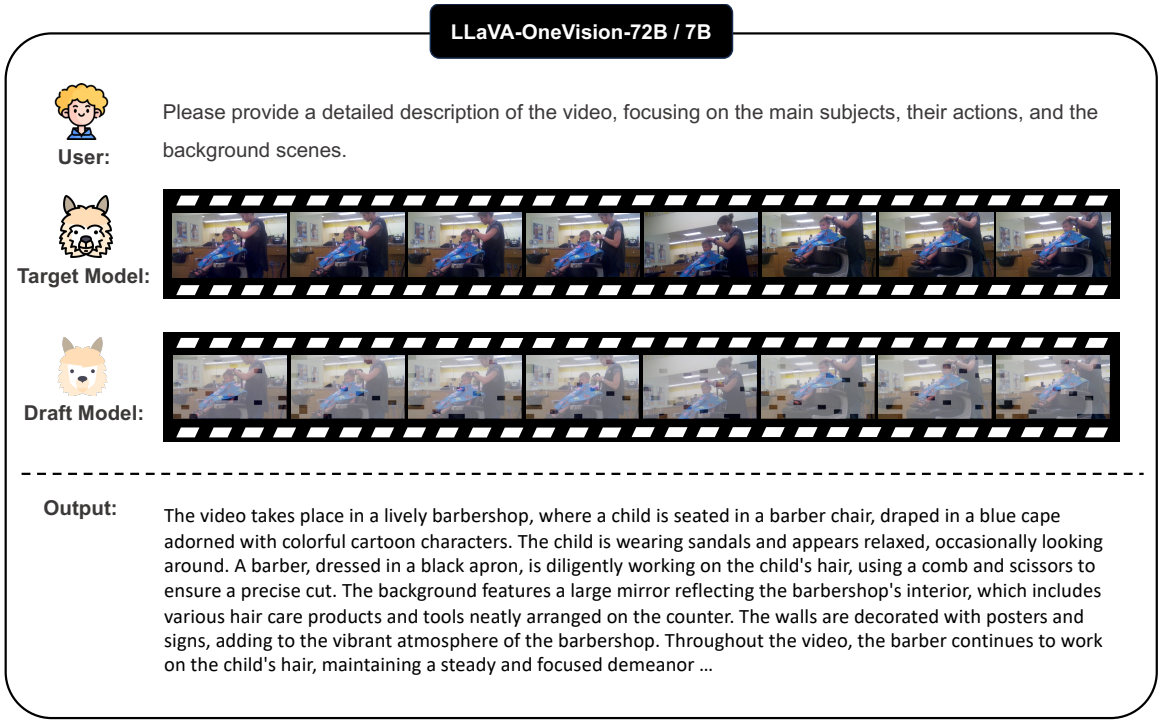

Figure 10: Visualization of SpecVLM on VideoDetailCaption using LLaVA-OneVision-72B / 7B.

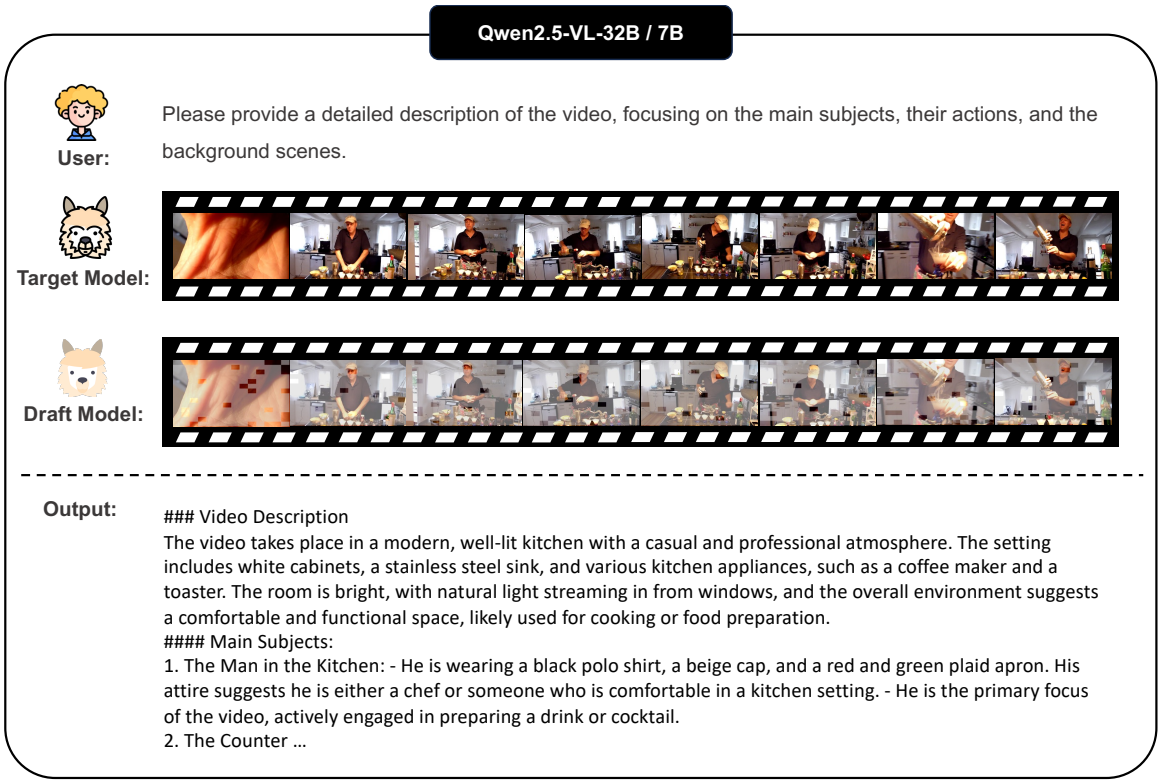

Figure 11: Visualization of SpecVLM on VideoDetailCaption using Qwen2.5-VL-32B / 7B.