

HiViS: Hiding Visual Tokens from the Drafter for Speculative Decoding in Vision-Language Models

Zhinan Xie^{1,2} Peisong Wang^{1,2,3} Shuang Qiu⁴ Jian Cheng^{1,2,3}

¹C²DL, Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³AIRIA ⁴City University of Hong Kong

Abstract

*Speculative decoding has proven effective for accelerating inference in Large Language Models (LLMs), yet its extension to Vision-Language Models (VLMs) remains limited by the computational burden and semantic inconsistency introduced by visual tokens. Recent studies reveal that visual tokens in large VLMs are highly redundant, and most of them can be removed without compromising generation quality. Motivated by this observation, we propose HiViS (**H**iding **V**isual **T**okens from the **D**rafter for **S**peculative **D**ecoding in **V**ision-**L**anguage **M**odels), a framework that utilizes the target VLM as a semantic fusion model, allowing the drafter to obtain visual information without explicitly processing visual tokens, ensuring that the drafter's prefill sequence length matches that of the textual tokens. Furthermore, HiViS employs a time-step-aware aligned training scheme that allows the drafter to autonomously propagate and refine instructive visual-textual semantics during independent drafting, guided by step-dependent bias-correction residuals. Extensive experiments across representative VLMs and benchmarks demonstrate that HiViS achieves significant improvements in average acceptance length and speedup ratio.*

1. Introduction

Autoregressive (AR) models form the backbone of modern generative AI. Large Language Models (LLMs) [1, 29] provide strong language understanding and generation capabilities, and Vision-Language Models (VLMs) [18, 22, 23] extend the same AR formulation to multimodal tasks such as image understanding and visual question answering. Despite their effectiveness, AR inference is inherently sequential because each token depends on all previously generated ones, causing latency to be constrained by memory bandwidth and communication in both LLMs and VLMs.

Speculative decoding [4, 16, 33] addresses the sequential

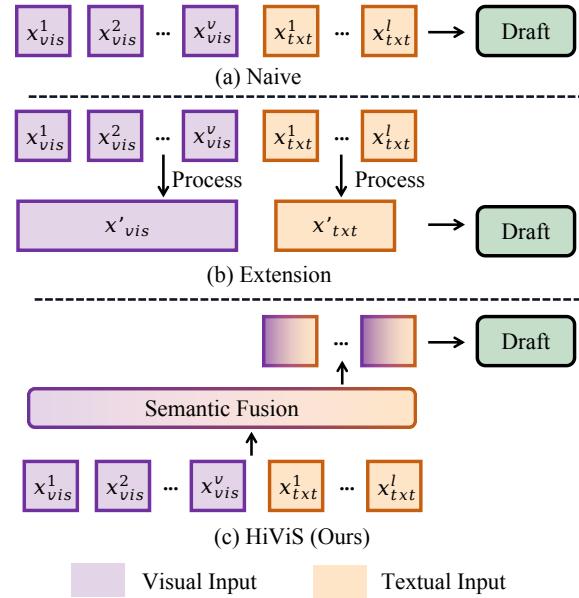


Figure 1. Input modalities for drafting.(a) Naive: feed both visual and textual tokens directly to the drafter. (b) Extensions: Additional module pre-processing visual and textual tokens before drafting. (c) HiViS: removes visual tokens and uses a semantic fusion module to enrich textual tokens with visual information.

inefficiency of autoregressive inference through a two-stage draft-verify process. A lightweight drafter first autoregressively generates candidate tokens, which the target model verifies in parallel within a single forward pass. Accepted tokens are retained, while rejected ones are resampled. This approach effectively reduces decoding steps while maintaining output quality. Although extensively explored in LLMs [3, 7, 9, 19, 27], its application to Vision-Language Models (VLMs) remains under-explored. This gap is non-trivial: modern VLMs inherit the large-scale architecture of LLMs and thus suffer from the same inference inefficiency, while also introducing additional complexity and fundamental obstacles due to the presence of visual tokens:

The capacity and semantic gap between drafter and target model. The visual tokens produced by the target VLM’s encoder are highly tailored to its own architecture and thus well aligned with target LLM’s semantic representation. This cross-modal alignment is achieved by the co-adaptation between the high capacity target LLM and its visual encoder. Although carrying rich multimodal semantics, these visual tokens are not naturally aligned with the representation space of a lightweight drafter. Feeding them directly into the drafter distorts the contextual signals, because the drafter has no capacity to deal with these cross-modal feature fusion. As a result, the mismatch between the drafter and target model leads to increased distributional discrepancy and lowers the acceptance rate.

Computational burden of long visual token sequences. Although the drafter remains lightweight to ensure efficiency, the requirement to handle long and complex visual token sequences increases its computational overhead. For instance, LLaVA-Next [23] produces over 2,000 visual tokens when processing high-resolution images, making visual inputs the dominant factor in sequence length and severely slowing down the drafter.

In this work, we propose **Hiding Visual Tokens from the Drafter for Speculative Decoding in Vision-Language Models (HiViS)**, a speculative decoding framework for VLMs that removes all visual tokens from the drafter yet still remaining visual information, as shown in Figure 1(c). We leverage the target VLM as a natural semantic fusion model for its cross-modality self-attention. This allows us to obtain visual injected text embeddings containing both visual and textual semantics, which are utilized as the drafter’s inputs, enabling the drafter to access multimodal information. While this scheme constructs a compact and semantically consistent KV-cache that accelerates decoding, the drafter cannot directly access updated visual semantics from the target VLM during independent drafting. To address this, we introduce a dynamic time-step aligned training mechanism in which a step-dependent bias correction residual is applied at each decoding step. This unified process adapts the visual injected text embeddings to the drafter and enables the drafter to propagate and update visual-textual semantics over multiple steps, enabling the drafter to operate autonomously without feedback from the target VLM.

In conclusion, our contributions are as follows:

- We introduce a new paradigm for VLM speculative decoding in which the drafter avoids directly processing visual tokens, with support from the target VLM.
- We propose a training scheme that enables the drafter to implicitly propagate visual and textual information during independent drafting, augmented with step-dependent bias-correction residuals to support inference.
- Extensive experiments on representative VLMs and benchmarks demonstrate consistent gains in acceptance

length and speedup (up to $3.15\times$), while losslessly preserving the target VLM’s output distribution.

2. Related Works

2.1. Speculative Decoding

Speculative decoding accelerates autoregressive inference by employing a lightweight drafter M_q to predict candidate tokens for the next λ steps, which are then verified in parallel by the target model M_p . During verification, M_p accepts consecutive tokens that match its distribution and rejects the first mismatch and its follows. The accepted prefix is retained, and M_p resamples the next token accordingly. If all candidates are accepted, M_p samples a new token from its original distribution. The number of accepted tokens per iteration, n , defines the effective inference length, allowing multiple tokens to be generated per forward pass with without sacrificing output quality.

Despite employing a lightweight drafter, prior studies [4, 16] demonstrate that performance critically depends on n , which is influenced by the drafter’s inference efficiency and its alignment with the target model. Among existing speculative decoding frameworks for LLMs, GLIDE[7] and EAGLE-2 [20] are representative examples. GLIDE[7] adopts a drafter with one self-attention and one cross-attention layer, and EAGLE-2 [20] uses a single-layer decoder tightly aligned with the target model. Both further employ tree-based drafting to raise acceptance rates, yielding larger speedups in practice. In our experiments, we adopt the tree-based drafting structure introduced in EAGLE-2 [20] to enable parallel candidate generation.

2.2. Large VLMs

We focus on large VLMs [22, 31, 32] that leverage pre-trained Large Language Models (LLMs) [6, 28, 29] as backbones, aligning visual representations with powerful language generation capabilities. Given an image-text pair, the vision encoder and multimodel projector extracts high-level visual embeddings $e_{\text{visual}} \in \mathbb{R}^{v \times d}$, which are projected into the textual embedding space, while textual inputs are embedded as $e_{\text{text}} \in \mathbb{R}^{l \times d}$. At each decoding step, the model samples the next token from $p(\cdot | e_{\text{visual}}, e_{\text{text}})$, enabling seamless reuse of strong LLM architectures while integrating visual semantics through lightweight projection or cross-modal alignment modules.

However, the large number of visual tokens in such models greatly inflates KV-cache size and sequence length, becoming a major bottleneck for efficient inference. Recent works alleviate this by compressing the KV-cache [12, 30], compressing visual tokens [39], or pruning [5]. To further improve inference efficiency without compromising output fidelity, we explore extending the lossless acceleration paradigm of speculative decoding to VLMs.

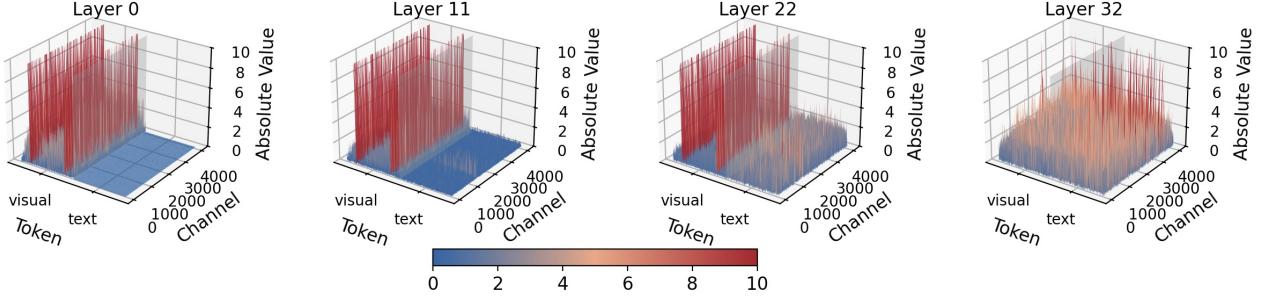


Figure 2. Visualization of hidden state magnitude distributions across layers. The gray divider marks the boundary between visual and text tokens.

3. Method

In this section, we first delve into how visual information and visual tokens affect the drafter in speculative decoding, which has inspired us to propose the HiViS framework. Then, we will introduce the HiViS framework in details.

3.1. Motivation

Typically, the inputs of VLMs include visual tokens and text tokens, which are transformed into embeddings before sending to the LLM backbone as follows:

$$\langle e_{\text{visual}}^1, e_{\text{visual}}^2, \dots, e_{\text{visual}}^v, e_{\text{text}}^1, e_{\text{text}}^2, \dots, e_{\text{text}}^l \rangle,$$

where $e_{\text{visual}} \in \mathbb{R}^{v \times d}$ and $e_{\text{text}} \in \mathbb{R}^{l \times d}$ represent the visual and text tokens, respectively.

Current speculative decoding methods for VLMs struggle with handling visual tokens: (1) Require the drafters to process visual tokens independently for alignment, which preserves the computational overhead of long visual sequences. (2) The lightweight drafter, limited in capacity, often fails to capture the key visual information emphasized by the target VLM, causing semantic misalignment. As shown in Figure 1, approaches like MSD [21] decouple visual and textual streams but still process both, while methods like IbED [15] introduce complex visual transformation before drafting, further increasing alignment difficulty. Consequently, both inefficiency and misalignment from visual tokens remain unresolved.

On the other hand, recent studies [37, 38] demonstrate that in VLMs [22, 23], prefill-stage attention guides aggressive visual token pruning with minimal performance loss. This indicates that only a small subset of visual tokens is essential and that key visual semantics are already embedded in the VLM’s hidden representations. Thus, the drafter can safely omit visual tokens and rely on the fused high-level features provided by the target VLM.

These challenges raise a natural question: *can the drafter avoid directly processing visual tokens while still accessing visual semantics?*

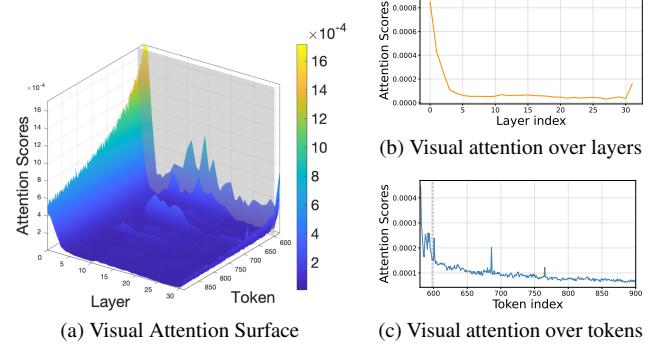


Figure 3. Visualization of vision attention across layers and tokens. (a) Mean attention from post-visual tokens to visual tokens across layers. The gray plane separates the instruction region from the generated text. (b) Attention variation curves obtained over layers. (c) Attention variation curves obtained over tokens.

3.2. Hiding Visual Tokens from the Drafter

3.2.1. Vision-to-Language Transition in Large VLMs

To answer the above question, we conduct a layerwise and token-level analysis of hidden states and attention patterns in LLaVA-1.5-7B [22] to investigate how visual information is represented.

The modality gap between visual and text tokens. We first assess the alignment between visual and textual features in target VLMs. To this end, we visualize the feature distribution of hidden state across layers in Figure 2, with the gray plane separating visual and text tokens. From Figure 2, it is easy to conclude that, although large-scale modality alignment pre-training is performed, the visual and textual representation spaces are not well aligned. Specifically, the visual feature distributions exhibit much larger magnitude and variance than the textual feature distributions, especially for early layers. As a result, visual tokens can merely provide auxiliary visual cues for text tokens, rather than forming an integrated context for language modeling.

The mismatch between target model’s visual abstraction level and draft model’s capacity. The modality gap presents a less critical challenge in target VLM, as the target LLM relies on multiple decoder layers to progressively integrate and refine visual information within the language space. This can be confirmed from the distribution transition across layers in Figure 2. In early layers, visual tokens exhibit larger magnitudes, reflecting strong visual influence. As depth increases, visual contributions diminish while textual representations strengthen. By the final layers, the magnitudes across modalities appear approximately balanced, with a tendency toward textual dominance, reflecting a shift from vision to language space representation. The visualization of attention intensity to visual tokens across layers in Figure 3 (a) and (b) also confirms that the target LLM progressively perceives visual information.

However, the draft model, which is designed with limited network capacity to ensure efficiency, lacks the necessary representational power to bridge this gap and interpret raw visual tokens directly. Feeding these raw visual tokens into the draft model would consequently impair its language modeling capacity and degrade performance. Therefore, a more effective strategy is to relieve the draft model from visual perception, while concentrating only on the language modeling ability.

The importance of text tokens gradually increases as decoding proceeds. Figure 3 visualizes textual to visual tokens attention across layers and decoding steps, with the gray plane separating instruction and generation regions. Attention to visual tokens drops sharply in initial layers and remains low afterward. The instruction region exhibits stronger visual attention, indicating that visual information is primarily integrated during context establishment, while later decoding depends only weakly on visual tokens.

3.2.2. Semantic Fusion and Drafter Workflow

The analyses reveal that the target VLM’s ability to perceive and understand visual tokens is enabled by its deep multi-layered architecture. However, the draft model does not have enough capacity to deal with both visual perception and contextual modeling. Thus our motivation is to remove the visual perception from the draft model.

However, we find that directly remove all visual information from the draft model is not a good choice. To further extend the observation, we conduct a variant-input experiment by varying the information provided to the drafter during training and inference. As shown in Table 1, including visual information from the target VLM while removing all visual tokens (here we denote this setting as visual injected text tokens, and will be described in the following section) not only eliminates redundant computation but also yields superior performance compared to using full visual-text inputs, where redundant visual information leads to suboptimal alignment.

Variant input for train & draft	average accept length
(a) text tokens only	2.16
(b) visual tokens & text tokens	3.04
(c) visual injected text tokens	3.28

Table 1. Drafters are trained on the same multimodal dataset and evaluated on LLaVA-Next-7B with ChartQA at temperature 0, with three input configurations: (a) text-only input, (b) visual & text tokens, and (c) visual injected text tokens.

mal alignment. Conversely, when the drafter receives only textual inputs without any visual information, performance degrades significantly. These results demonstrate that effective visual semantics are neither captured by raw visual tokens nor recoverable from text alone, but are instead best conveyed through the fused representations provided by the target VLM. In short, for the draft model in VLM, *the visual information is indispensable, but the visual tokens are not*. These observations support HiViS to operate without explicit visual tokens, with the help from the target VLM.

The overall framework. We introduce the HiViS framework in details. Our key insight is that visual semantics can be preserved without requiring the drafter to explicitly process visual tokens. The target VLM, serving as a semantic fusion model, provides cross-modality-enriched hidden states that are linearly combined with textual embeddings to form fused inputs for the drafter, relieving the draft model from visual perception while concentrating on language modeling.

As illustrated in Figure 4, the input to the target VLM M_p consists of two modalities: visual tokens and text tokens. The visual branch is obtained by passing the image through a vision encoder followed by a multimodal projector, producing a sequence of v visual embeddings $e_{\text{visual}}^1, \dots, e_{\text{visual}}^v$. In parallel, the text branch is formed by embedding the text tokens, yielding $e_{\text{text}}^1, \dots, e_{\text{text}}^l$.

Within the target VLM M_p , which typically consists of multiple decoder layers (denote the number as N), each decoder performs self-attention over both textual and visual tokens to achieve multimodal fusion. At decoding step $t > v$, the input to the n_{th} decoder layer is defined as $F_{n-1} = [f_{\text{visual}}^{1:v}, f_{\text{text}}^{1:t-v}] \in \mathbb{R}^{t \times d}$, where $F_0 = [e_{\text{visual}}^{1:v}, e_{\text{text}}^{1:t-v}]$. We employ the final-layer outputs $F_N^{v+1:l}$ as the fused features $f_{\text{fused}}^{1:l}$ provided to the drafter M_q . In addition, the drafter requires explicit textual inputs $e_{\text{text}}^{1:l} \in \mathbb{R}^{l \times d}$. We form the final visual injected text embeddings by a linear combination of the fused features and the causally shifted text embeddings $\hat{e}_{\text{text}}^{2:l+1} = f_{\text{fused}}^{1:l}w_1 + e_{\text{text}}^{2:l+1}w_2 \in \mathbb{R}^{l \times d}$, where $w_1, w_2 \in \mathbb{R}^{d \times d}$ are learnable parameters.

Algorithm 1 shows a detailed utilization of the visual injected text embedding within the draft-verify cycle. During the draft stage, since the drafter cannot access the target VLM’s fused features, at draft step i it substitutes its own

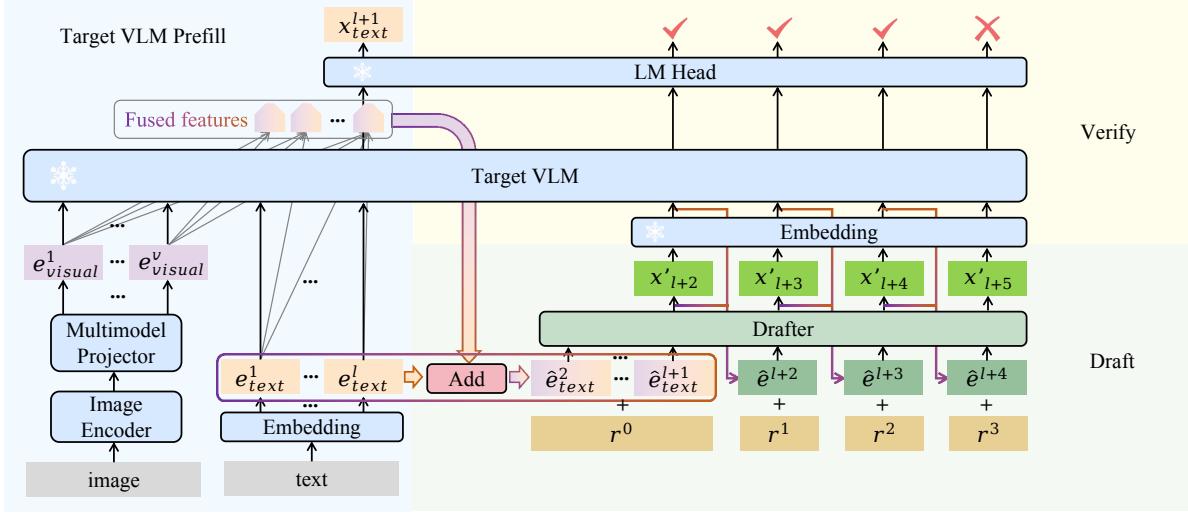


Figure 4. Overall framework of HiViS during the draft-verify process. The target VLM first performs the multimodal prefill to generate fused features f_{fused} from both visual embeddings e_{visual} and textual embeddings e_{text} . These fused features are used to construct the visual injected text embeddings \hat{e}_{text} that serve as inputs to the drafter. During the draft stage, the drafter generates multiple candidate, each step refined by a step-dependent bias-correction residual r . In the verify stage, the target VLM evaluates all candidates in parallel and accepts the consecutive correct sequence.

Algorithm 1 HiViS: Residual-Aided Drafting and Verification Process

Inputs: Target VLM M_p , Drafter M_q , $e_{\text{visual}} \in \mathbb{R}^{v \times d}$, $e_{\text{text}} \in \mathbb{R}^{l \times d}$, $f_{\text{fused}}^{1:l}, e_{\text{text}}^{1:l} \leftarrow \text{Prefill}(M_p; e_{\text{visual}}^{1:v}, e_{\text{text}}^{1:l})$

▷ Prefill the target VLM and obtain fused textual representations carrying visual semantics.

- 1: $\hat{e}_{\text{text}}^{2:l+1} \leftarrow f_{\text{fused}}^{1:l} w_1 + e_{\text{text}}^{2:l+1} w_2$
▷ Semantic Fusion: text embeddings fuse visual-aware fused feature forming visual injected text embedding.
- 2: $\text{prefix} \leftarrow \hat{e}_{\text{text}}^{2:l} + \text{Expend}(r^0 w), \hat{e}^{l+1} \leftarrow \hat{e}_{\text{text}}^{l+1}$
▷ Build prefix with initial bias correction residual r^0 .
- 3: **for** $i = 1$ **to** λ **do**
▷ Step-wise drafting: lightweight drafter autoregressively generates fused features f_{new} and tokens e_{new} .
- 4: $f'_{l+i} \leftarrow M_q(\text{prefix}, \hat{e}^{l+1:l+i} + r^{0:i-1} w)$
- 5: $x'_{l+i+1} \leftarrow \text{LM Head}(f'_{l+i})$
- 6: $\hat{e}^{l+i+1} \leftarrow f'_{l+i} w_1 + \text{Embed}(x'_{l+i+1}) w_2$
▷ Combine fused features and textual embeddings to propagate implicit visual semantics.
- 7: **end for**
- 8: $x'_{l+2:l+1+n} \leftarrow \text{Parallel Verify}(M_p, x'_{l+2:l+1+\lambda})$
▷ Parallel verification by target VLM.
- 9: **return** $e_{\text{text}} \leftarrow e_{\text{text}}^{1:l} \oplus \text{Embed}(x'_{l+1:l+1+n})$

hidden state from the previous decoder layer, f'_{l+i} , to construct the current representation \hat{e}^{l+i+1} . The input is then formed as $\hat{e}^{l+i+1} + r^i w$, where r is the learnable bias correction residual specifically tailored for the draft phase to alleviate the accumulation of error arising from the visual

injected text embedding over time, with $w \in \mathbb{R}^{d \times d}$. The detailed design and training of this residual mechanism are described in the following section.

3.3. Time-step-aware Residual Aligned Training

The HiViS framework can overcome the visual token hurdle in the draft model, however, mismatch still exists between the drafter and the target model. The main reason for this mismatch is that during drafting stage, the drafter cannot obtain the target VLM’s latest visual-textual semantic updates. We will provide a detailed elaboration of the problem and the corresponding solution.

Step-dependent bias-correction residuals. HiViS conditions the drafter on visual injected text embeddings whose fused-feature component is supplied by the target VLM. However, this information is not available during drafting. Instead, the hidden state from the previous step of the draft model is used, i.e., the f'_t in Figure 5, which causes the mismatch between the drafter and the target model. To fill this gap, we design a step-dependent bias-correction residuals (the r^i in Figure 5), which are linearly added to the visual injected text embeddings at each draft decoding step. These residuals progressively adjust the visual injected text embeddings to support stable semantic propagation over independent drafting steps.

To train these step-dependent bias-correction residual terms, HiViS performs N step inference simulation to mimic the drafter’s independent decoding process, as illustrated in Figure 5 (with $N = 5$) [36]. At the starting timestep t for each drafting round, the drafter receives the

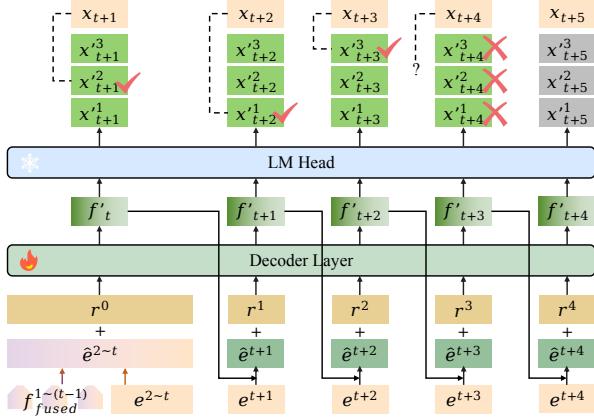


Figure 5. Training architecture of HiViS. e are input embeddings, f_{fused} and x are fused feature and tokens from the target VLM, while f' denotes the drafter-predicted hidden states and x'^k the corresponding top- K token candidates drawn from its output distribution. Correct predictions are marked with \checkmark , errors with \times , and gray tokens denote discarded candidates.

embeddings up to t , denoted as $[e^1, e^2, \dots, e^t]$. The initial visual injected text embedding is constructed as $\hat{e}^{2:t} = f_{\text{fused}}^{1:(t-1)}w_1 + e^{2:t}w_2$, where $f_{\text{fused}}^{1:(t-1)}$ represents the fused hidden states provided by the target VLM. At the first simulation step, the initial residual r^0 is applied directly to all visual injected text embeddings to better align with the drafter’s processing. Each subsequent simulation step i has its own residual r^{i-1} , which is linearly combined with the input embedding before being fed into the drafter. These residuals progressively refine the representations and compensate for deviations that accumulate across multiple simulation steps. As shown in Figure 5, in the first forward pass ($i = 1$), the drafter predicts its hidden state as:

$$f'_t = M_q(\hat{e}^{2:t} + r^0 w). \quad (1)$$

Subsequently, the i_{th} forward pass ($1 < i \leq N$) can be generalized as:

$$f'_{t+i} = M_q(\hat{e}^{2:t} + r^0 w, \hat{e}^{t+1} + r^1 w, \dots, \hat{e}^{t+i-1} + r^{i-1} w). \quad (2)$$

Two-stage training with early stopping Furthermore, we propose a dynamic top- K filtering strategy with early stopping to keep within-capacity examples while removing consistently failed hard token samples, enhancing training efficiency and semantic consistency across extended generation. Specifically, our training contains two stages, at the stage-1, we only train the drafter with one step, and at the stage-2, we train it for $N > 1$ steps. In this way, the stage-1 training establishes a solid foundation for the drafter’s predictive capability, while the stage-2 both simulates the draft token tree construction process of actual drafting and performs dynamic data filtering.

For stage-2, inspired by knowledge distillation [34], the drafter outputs a probability distribution at each step, from which the top- K tokens are selected as candidates. We then apply a dynamic filtering criterion:

$$x'_{t+i} \in \{x'^1_{t+i}, \dots, x'^k_{t+i}, \dots, x'^K_{t+i}\} \quad (3)$$

where x'^k_{t+i} denotes the k_{th} highest probability predicted by the drafter. If the target VLM generated ground-truth token x_{t+i} appears in the drafter’s top- K predictions, the example at i_{th} forward pass (timestep $t+i-1$) is kept for optimization. Otherwise, the simulation is terminated early and discarded to avoid wasting effort on consistently mispredicted sequences. As shown in Figure 5 with $N = 5$, the process stops at the 4_{th} forward pass for the candidate tokens no longer cover the target VLM’s prediction, and thus the 5_{th} forward pass is excluded from training.

Training loss. The overall loss consists of three components, weighted as:

$$L = L_{\text{fus}} + \beta L_{\text{cls}} + \gamma L_{\text{topK}}, \quad (4)$$

where

$$L_{\text{fus}} = \text{Smooth-L1}(f_{\text{fused}}^t, f'_t),$$

$$L_{\text{cls}} = \text{CrossEntropy}(p_t, q_t),$$

$$L_{\text{topK}} = - \sum_{x \in \Omega_t} q_t(x) \log p_t(x).$$

Here, f_{fused}^t denotes the target VLM’s fused feature at step t and f'_t the drafter’s hidden state, p_t and q_t are the predicted probability distributions of target VLM and drafter, Ω_t is the set of top- K tokens under the target at step t . Accordingly, L_{fus} supervises representation alignment supporting propagation of visual-textual semantics, L_{cls} enforces distributional alignment by matching next-token logits, and L_{topK} emphasizes the ranking loss [36] for augmenting the alignment between the top-ranked tokens.

4. Experiment

Models and Tasks: To assess the effectiveness and generality of HiViS, we evaluate three representative open-source VLMs: LLaVA-1.5 (7B, 13B) [22], LLaVA-Next (7B, 13B) [23], and Qwen2.5-VL-7B [2]. Experiments are conducted on standard VQA benchmarks, including ChartQA [25], VQAv2 [10], ScienceQA [24], TextVQA [26], MME [8], MM-Vet [35], SEED-Bench [17], and GQA [13], covering diverse image understanding and reasoning tasks. Each benchmark is evaluated on an 80-sample subset.

Baselines: We compare HiViS with four baselines. EAGLE-2 [20], a one-decoder-layer speculative decoding framework for LLMs, is reimplemented and retrained on the multimodal dataset. For VLM-specific baselines, we evaluate MSD [21], ViSpec [14], and Dream [11] using their

Table 2. Speedup ratio (SR) and average acceptance length (τ) across benchmarks.

Model	Methods	ChartQA SR	ChartQA τ	VQAv2 SR	VQAv2 τ	ScienceQA SR	ScienceQA τ	TextVQA SR	TextVQA τ	MME SR	MME τ	MMVet SR	MMVet τ	SEED-Bench SR	SEED-Bench τ	GQA SR	GQA τ	Avg SR	Avg τ
T = 0																			
LLaVA-1.5-7B	EAGLE-2	1.56×	3.06	1.95×	4.62	1.44×	2.97	1.88×	3.38	1.77×	3.72	1.89×	3.71	2.00×	4.64	1.83×	4.39	1.79×	3.81
	MSD	2.06×	4.53	2.01×	5.06	2.03×	4.27	2.09×	4.22	2.02×	4.74	2.06×	4.38	2.05×	4.97	1.99×	5.16	2.04×	4.67
	HiViS	2.37×	5.13	2.28 ×	5.78	2.26 ×	5.12	2.55 ×	4.95	2.44 ×	5.44	2.42 ×	5.03	2.29 ×	5.56	2.25 ×	5.95	2.36 ×	5.37
LLaVA-1.5-13B	EAGLE-2	1.73×	3.18	2.05×	4.78	1.57×	2.10	1.95×	3.31	1.84×	3.54	2.04×	3.75	2.07×	4.62	1.92×	4.50	1.90×	3.72
	MSD	2.15×	4.28	2.07×	4.59	2.02×	3.87	2.12×	3.92	2.07×	3.82	2.14×	4.08	2.11×	4.70	2.00×	4.58	2.09×	4.23
	HiViS	2.43×	5.24	2.20 ×	5.44	2.17 ×	4.76	2.50 ×	4.69	2.35 ×	5.04	2.52 ×	5.05	2.28 ×	5.46	2.15 ×	5.51	2.33 ×	5.15
LLaVA-Next-7B	EAGLE-2	1.33×	3.04	1.59×	4.52	1.47×	3.00	1.43×	3.04	1.43×	3.35	1.42×	2.81	1.36×	4.33	1.39×	4.36	1.43×	3.56
	Dream	1.28×	3.62	1.34×	3.21	1.39×	3.65	1.06×	2.72	1.20×	3.30	1.16×	2.82	1.36×	4.35	1.32×	4.28	1.26×	3.49
	ViSpec	1.66×	4.36	1.53×	4.17	1.78×	4.19	1.99×	4.46	1.59×	4.03	1.70×	4.17	1.58×	4.33	1.38×	4.18	1.65×	4.24
LLaVA-Next-13B	EAGLE-2	1.43×	3.19	1.63×	4.55	1.56×	3.06	1.52×	2.98	1.53×	3.39	1.54×	3.23	1.28×	4.33	1.31×	4.38	1.48×	3.64
	ViSpec	1.76×	4.47	1.61×	4.25	1.87×	4.03	2.03×	4.25	1.72×	4.10	1.76×	3.96	1.74×	4.62	1.34×	4.26	1.73×	4.24
	HiViS	1.88×	5.05	1.79 ×	5.48	2.14 ×	4.80	2.33 ×	4.90	1.89 ×	4.70	1.97 ×	4.67	1.85 ×	5.25	1.43 ×	5.26	1.91 ×	5.01
Qwen2.5-VL-7B	EAGLE-2	1.33×	2.49	2.09×	4.40	1.41×	2.62	2.31×	2.78	1.60×	2.96	1.52×	2.57	1.80×	3.79	1.96×	4.14	1.75×	3.22
	ViSpec	1.81×	3.90	1.90×	3.73	1.86×	3.71	2.87×	3.60	1.75×	3.46	1.75×	3.44	1.69×	3.68	1.92×	3.87	1.94×	3.67
	HiViS	2.04×	4.56	2.28 ×	4.92	2.13 ×	4.34	3.15 ×	3.95	1.98 ×	3.98	2.00 ×	4.01	2.00 ×	4.54	2.21 ×	4.84	2.22 ×	4.39
T = 1																			
LLaVA-1.5-7B	EAGLE-2	1.42×	2.66	1.77×	3.68	1.29×	2.58	1.38×	2.49	1.49×	2.85	1.53×	2.82	1.77×	3.51	1.71×	3.64	1.55×	3.03
	MSD	1.70×	3.29	1.79×	3.86	1.59×	3.27	1.51×	2.96	1.65×	3.36	1.66×	3.23	1.76×	3.84	1.82×	4.00	1.69×	3.48
	HiViS	1.95×	3.60	1.97 ×	3.99	1.84 ×	3.63	1.71 ×	3.32	1.89 ×	3.61	1.90 ×	3.48	1.95 ×	3.92	2.00 ×	4.34	1.90 ×	3.74
LLaVA-1.5-13B	EAGLE-2	1.57×	2.80	1.83×	3.74	1.52×	2.65	1.58×	2.65	1.66×	2.91	1.69×	2.91	1.85×	3.71	1.86×	3.74	1.70×	3.14
	MSD	1.86×	3.40	1.92×	3.72	1.71×	3.18	1.66×	3.00	1.76×	3.24	1.80×	3.15	1.90×	3.71	1.85×	3.86	1.81×	3.41
	HiViS	2.04×	3.70	1.96 ×	4.18	1.92 ×	3.49	1.91 ×	3.12	2.02 ×	3.78	2.00 ×	3.58	2.02 ×	4.07	1.98 ×	4.11	1.98 ×	3.75
LLaVA-Next-7B	EAGLE-2	1.18×	2.63	1.43×	3.46	1.20×	2.49	1.07×	2.30	1.26×	2.63	1.23×	2.65	1.21×	3.23	1.24×	3.42	1.23×	2.85
	Dream	1.18×	3.31	1.25×	3.84	1.33×	3.34	0.96×	2.51	1.14×	3.1	1.11×	2.87	1.26×	3.01	1.22×	4.03	1.18×	3.25
	ViSpec	1.46×	3.49	1.42×	3.37	1.40×	3.18	1.35×	3.04	1.36×	3.15	1.40×	3.11	1.31×	3.40	1.25×	3.38	1.37×	3.27
LLaVA-Next-13B	EAGLE-2	1.27×	2.71	1.51×	3.47	1.33×	2.63	1.26×	2.46	1.41×	2.87	1.35×	2.72	1.21×	3.29	1.25×	3.52	1.32×	2.96
	ViSpec	1.57×	3.58	1.53×	3.44	1.64×	3.21	1.52×	3.21	1.59×	3.30	1.53×	3.24	1.49×	3.50	1.24×	3.49	1.51×	3.37
	HiViS	1.76×	3.82	1.74 ×	3.99	1.75 ×	3.48	1.68 ×	3.37	1.68 ×	3.60	1.66 ×	3.54	1.61 ×	3.85	1.25 ×	3.89	1.64 ×	3.69
Qwen2.5-VL-7B	EAGLE-2	1.28×	2.53	1.72×	3.34	1.33×	2.50	1.87×	2.37	1.42×	2.63	1.31×	2.35	1.64×	3.24	1.71×	3.33	1.54×	2.79
	ViSpec	1.72×	3.88	1.63×	3.10	1.72×	3.45	2.27×	2.96	1.51×	2.97	1.54×	3.06	1.55×	3.41	1.62×	3.20	1.70×	3.25
	HiViS	1.89×	4.38	1.86 ×	3.63	1.97 ×	3.97	2.39 ×	3.08	1.70 ×	3.27	1.67 ×	3.40	1.79 ×	3.72	1.86 ×	3.84	1.89 ×	3.66

publicly released model weights. MSD employs a single-layer decoder drafter that decouples visual and textual processing but still relies on full multimodal inputs. ViSpec uses a single-layer drafter that explicitly compresses visual tokens during inference, while Dream adopts a two-layer decoder with one cross-attention module and applies visual token pruning to accelerate generation.

Metrics: As speculative decoding is central to this work, evaluation follows standard protocols and focuses on inference efficiency, using two primary metrics:

- **Speedup Ratio (SR).** The throughput ratio of speculative decoding to standard autoregressive generation, measured in tokens per second and indicating practical speedup.
- **Average Acceptance Length (τ).** The average number of drafter tokens accepted per verification step. Larger τ indicates stronger alignment with the target VLM and higher decoding efficiency.

Training Setup: Consistent with the data configuration used in MSD [21], we mix 68,000 ShareGPT dialogues as the text-only dataset and 68,000 samples from LLaVA-

Mix665k as the multimodal dataset. Only textual embeddings are retained, alleviating training overhead. Since many multimodal samples contain single-token answers, which is unsuitable for time-step aligned training, we filter out entries with answers shorter than five words.

During training, we cap the draft length at $N = 4$ to efficiently learn the bias-correction residuals r^0 - r^3 . At inference time, if drafting exceeds four steps, we reuse r^3 for all later steps. We use $K = 5$ top candidates per step and set loss coefficients to $\beta = 0.1$ and $\gamma = 1$. All remaining hyperparameters follow EAGLE-2 [20].

Inference Setup: Experiments are conducted on a single A800 GPU with batch size 1. During drafting, we expand a top-10 draft tree to depth $\lambda = 6$ and retain the top-60 paths ranked by cumulative probability.

4.1. Results

Table 2 summarizes the speedup ratio and average acceptance length across multiple VLMs and tasks. Even without any explicit visual inputs during training or inference,

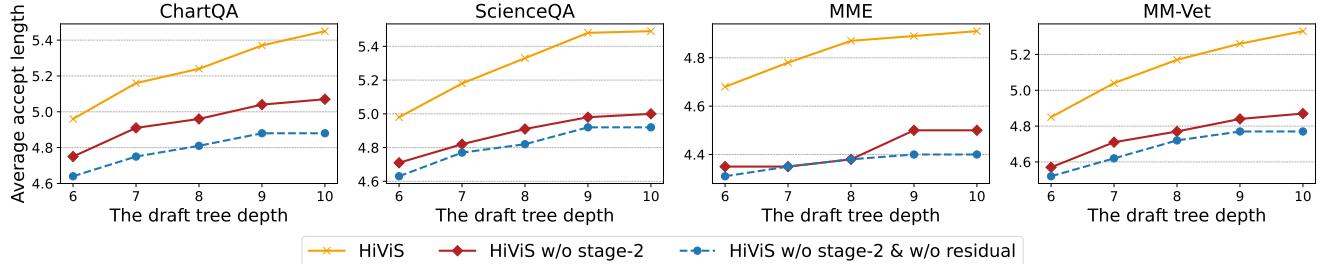


Figure 6. Ablation study of HiViS on four tasks evaluated on LLaVA-Next-7B under different draft tree depths (6-10). The comparison includes: (1) full HiViS, (2) HiViS without stage-2 training, and (3) HiViS without both stage-2 and bias-correction residuals.

HiViS consistently attains higher speedup ratios than all baselines. At temperature $T = 0$ (greedy sampling), HiViS achieves up to a $3.15\times$ speedup on Qwen2.5-VL-7B for TextVQA, exceeding ViSpec ($2.87\times$) and EAGLE-2 ($2.31\times$). When sampling with $T = 1$, HiViS maintains clear advantages, achieving speedups between $1.48\times$ and $2.39\times$, along with higher acceptance lengths, for example 4.38 on ChartQA with Qwen2.5-VL-7B, compared with 3.88 for ViSpec and 2.53 for EAGLE-2. Since ViSpec adopts additional system prompt, task-specific instructions, and different draft parameters, we further evaluate HiViS under their setting without additional training. Even so, HiViS still surpasses ViSpec across all tasks in both speedup ratio and average acceptance length. Full results are included in the Supplementary Material.

These results show that drafters for VLM can function effectively without explicit visual tokens, confirming the efficiency and robustness of our framework.

4.2. Ablation Studies

We conduct studies on LLaVA-Next-7B at temperature 0, examining how visual tokens, training stages and dataset composition affect the drafter, measured by average acceptance length τ .

Effect of visual tokens. We vary visual-token input and training configurations in stage-1 as shown in Table 3. Removing visual tokens already improves performance when trained on the multimodal dataset, and adding text-only data plus bias correction residual training yields a further boost. These results indicate that explicit visual tokens are unnecessary for effective speculative decoding.

Effect of time-step aligned training and bias correction residual. Figure 6 reports average acceptance length for draft tree depths 6 to 10. Across all tasks, HiViS without stage-2 training or residuals performs the worst. Adding residuals yields clear improvement, indicating that the initial r^0 already helps stabilize independent drafting. After stage-2 refinement, acceptance length becomes the highest and continues to grow with depth, showing that the residual-based, time-step aligned training strengthens drafter’s alignment with the target VLM over longer drafting sequences.

Table 3. Stage-1 with different input sequences (“txt” for text, “vis” for visual) and training skills.

Input tokens	Training	MME		MM-Vet	
		SR	τ	SR	τ
vis & txt	multimodal data	1.33x	3.35	1.42x	2.81
txt	multimodal data	1.59x	3.76	1.59x	3.16
txt	+ text data & residual	1.82x	4.35	1.87x	4.57

Table 4. Effect of stage-1 training dataset composition on average acceptance length. Drafters trained with both multimodal and text-only data achieve the best performance across tasks.

Stage-1 Training Dataset	MME	MM-Vet
Multimodal only	3.86	3.23
Text only	3.85	4.22
Multimodal + text	4.35	4.57

Effect of dataset composition. We analyze the impact of stage-1 dataset composition across different tasks. As shown in Table 4, the performance varies when trained on multimodal or text-only data, reflecting their distinct contributions to different task types. Combining both datasets consistently achieves the best results, suggesting that the complementary strengths of multimodal and textual supervision enhance the drafter’s robustness and generalization.

5. Conclusion

We present HiViS, a speculative decoding framework for VLMs that removes all visual tokens from the drafter while preserving multimodal semantics by using the target VLM as a natural semantic fusion module. HiViS further strengthens the drafter via time-step aligned training with step-dependent residuals. Across multiple VLMs and benchmarks, HiViS achieves the highest acceleration while maintaining strong alignment with the target VLM, demonstrating that speculative decoding for VLMs can be performed effectively without explicit visual tokens and paving the way for more lightweight drafter designs.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6
- [3] Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024. 1
- [4] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023. 1, 2
- [5] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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, 2024. 2
- [6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhang-hao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yong-hao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2
- [7] Cunxiao Du, Jing Jiang, Xu Yuanchen, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, and Yang You. Glide with a cape: a low-hassle method to accelerate speculative decoding. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024. 1, 2
- [8] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. 6
- [9] Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*, 2024. 1
- [10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 6
- [11] Yunhai Hu, Tianhua Xia, Zining Liu, Rahul Raman, Xingyu Liu, Bo Bao, Eric Sather, Vithursan Thangarasa, and Sai Qian Zhang. Dream: Drafting with refined target features and entropy-adaptive cross-attention fusion for multimodal speculative decoding. *arXiv preprint arXiv:2505.19201*, 2025. 6
- [12] Wenzuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, Yao Hu, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. *arXiv preprint arXiv:2412.00876*, 2024. 2
- [13] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 6
- [14] Jialiang Kang, Han Shu, Wenshuo Li, Yingjie Zhai, and Xinghao Chen. Vispec: Accelerating vision-language models with vision-aware speculative decoding. In *Annual Conference on Neural Information Processing Systems*, 2025. 6
- [15] Minjae Lee, Wonjun Kang, Byeongkeun Ahn, Christian Classen, Minghao Yan, Hyung Il Koo, and Kangwook Lee. In-batch ensemble drafting: Robust speculative decoding for LVLMs. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. 3
- [16] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023. 1, 2
- [17] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 6
- [18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1
- [19] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning*, 2024. 1
- [20] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Empirical Methods in Natural Language Processing*, 2024. 2, 6, 7
- [21] Luxi Lin, Zhihang Lin, Zhanpeng Zeng, and Rongrong Ji. Speculative decoding reimaged for multimodal large language models. *arXiv preprint arXiv:2505.14260*, 2025. 3, 6, 7
- [22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2, 3, 6
- [23] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 1, 2, 3, 6
- [24] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 6
- [25] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 6

- [26] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 6
- [27] Hanshi Sun, Zhuoming Chen, Xinyu Yang, Yuandong Tian, and Beidi Chen. Triforce: Lossless acceleration of long sequence generation with hierarchical speculative decoding. *arXiv preprint arXiv:2404.11912*, 2024. 1
- [28] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2, 2024. 2
- [29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2
- [30] Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VI-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. *arXiv preprint arXiv:2410.23317*, 2024. 2
- [31] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2
- [32] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2
- [33] Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925, Singapore, 2023. Association for Computational Linguistics. 1
- [34] Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024. 6
- [35] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 6
- [36] Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. Learning harmonized representations for speculative sampling. *arXiv preprint arXiv:2408.15766*, 2024. 5, 6
- [37] Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv e-prints*, pages arXiv–2412, 2024. 3
- [38] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. *arXiv preprint arXiv:2412.01818*, 2025. 3
- [39] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-mini: Efficient image and video large multimodal models with one vision token. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

6. Evaluation Under ViSpec Settings

we additionally evaluate HiViS and our reproduced EAGLE-2 under the exact settings used by ViSpec. Following their protocol, we prepend the same system prompt: *A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human’s questions.* and include all task-specific instructions provided in the dataset. During speculative decoding, we adopt ViSpec’s draft token tree configuration, using a total of 30 draft tokens, a tree depth of 4, and 8 nodes selected during each expansion step. Under this evaluation setup, without any additional training, HiViS still consistently surpasses ViSpec in both speedup ratio and average acceptance length across all tested tasks. Full results are reported in Table 5.

Table 5. Speedup ratio (SR) and average acceptance length (τ) across benchmarks under ViSpec’s settings.

Model	Methods	ChartQA SR	ChartQA τ	VQAv2 SR	VQAv2 τ	ScienceQA SR	ScienceQA τ	TextVQA SR	TextVQA τ	MME SR	MME τ	MMVet SR	MMVet τ	SEED-Bench SR	SEED-Bench τ	GQA SR	GQA τ	Avg SR	Avg τ
T = 0																			
LLaVA-Next-7B	EAGLE-2	1.37 \times	2.75	1.94 \times	3.66	1.52 \times	2.80	1.52 \times	2.96	1.53 \times	2.91	1.50 \times	2.80	1.67 \times	3.64	1.55 \times	3.56	1.58 \times	3.14
	ViSpec	1.75 \times	3.98	2.02 \times	3.82	2.36 \times	3.87	2.10 \times	3.99	1.84 \times	3.80	1.81 \times	3.81	1.78 \times	4.08	1.63 \times	3.86	1.91 \times	3.90
	HiViS	1.80 \times	4.03	2.19 \times	4.34	2.52 \times	4.05	2.19 \times	4.07	1.89 \times	3.81	1.90 \times	3.96	1.85 \times	4.17	1.72 \times	4.15	2.01 \times	4.07
LLaVA-Next-13B	EAGLE-2	1.50 \times	2.78	1.80 \times	3.67	1.69 \times	2.80	1.63 \times	2.08	1.66 \times	2.92	1.63 \times	2.82	1.75 \times	3.47	1.72 \times	3.61	1.67 \times	3.02
	ViSpec	1.84 \times	3.91	2.47 \times	3.88	2.39 \times	3.71	2.16 \times	3.78	2.00 \times	3.83	1.93 \times	3.66	1.94 \times	4.08	1.83 \times	3.93	2.07 \times	3.85
	HiViS	1.88 \times	4.01	2.61 \times	4.27	2.56 \times	4.00	2.32 \times	4.03	2.07 \times	3.93	2.02 \times	3.91	1.97 \times	4.13	1.87 \times	4.22	2.16 \times	4.06
Qwen2.5-VL-7B	EAGLE-2	1.46 \times	2.44	2.00 \times	3.47	1.57 \times	2.57	2.34 \times	2.44	1.58 \times	2.54	1.53 \times	2.37	1.71 \times	3.06	1.99 \times	3.43	1.77 \times	2.79
	ViSpec	1.85 \times	3.56	1.88 \times	3.22	1.87 \times	3.17	2.94 \times	3.22	1.80 \times	3.09	1.78 \times	3.08	1.74 \times	3.25	1.93 \times	3.30	1.97 \times	3.24
	HiViS	1.97 \times	3.85	2.13 \times	3.78	2.04 \times	3.47	3.04 \times	3.34	1.89 \times	3.25	1.89 \times	3.30	1.85 \times	3.51	2.15 \times	3.74	2.12 \times	3.53
T = 1																			
LLaVA-Next-7B	EAGLE-2	1.22 \times	2.34	1.48 \times	2.92	1.30 \times	2.33	1.21 \times	2.50	1.31 \times	2.38	1.30 \times	2.46	1.44 \times	2.80	1.42 \times	2.79	1.34 \times	2.57
	ViSpec	1.45 \times	3.18	1.82 \times	3.07	1.93 \times	2.96	1.50 \times	2.84	1.57 \times	2.99	1.51 \times	3.01	1.53 \times	3.15	1.43 \times	3.13	1.59 \times	3.04
	HiViS	1.60 \times	3.30	1.88 \times	3.26	2.14 \times	3.06	1.62 \times	2.84	1.63 \times	3.04	1.60 \times	3.06	1.55 \times	3.16	1.50 \times	3.26	1.69 \times	3.12
LLaVA-Next-13B	EAGLE-2	1.42 \times	2.50	1.60 \times	3.01	1.56 \times	2.29	1.42 \times	2.62	1.51 \times	2.50	1.50 \times	2.47	1.58 \times	2.85	1.47 \times	2.96	1.51 \times	2.65
	ViSpec	1.69 \times	3.28	2.14 \times	3.19	1.85 \times	2.94	1.73 \times	2.97	1.73 \times	3.09	1.77 \times	3.07	1.69 \times	3.23	1.64 \times	3.17	1.78 \times	3.12
	HiViS	1.76 \times	3.39	2.28 \times	3.40	2.16 \times	3.13	1.83 \times	3.07	1.78 \times	3.16	1.81 \times	3.20	1.89 \times	3.29	1.75 \times	3.37	1.91 \times	3.25
Qwen2.5-VL-7B	EAGLE-2	1.43 \times	2.42	1.63 \times	2.64	1.39 \times	2.28	1.79 \times	1.99	1.35 \times	2.14	1.33 \times	2.04	1.40 \times	2.41	1.61 \times	2.68	1.49 \times	2.33
	ViSpec	1.78 \times	3.41	1.57 \times	2.65	1.57 \times	2.67	2.15 \times	2.44	1.51 \times	2.47	1.54 \times	2.50	1.39 \times	2.54	1.57 \times	2.63	1.64 \times	2.66
	HiViS	1.89 \times	3.61	1.74 \times	2.92	1.73 \times	2.88	2.19 \times	2.47	1.59 \times	2.59	1.63 \times	2.65	1.43 \times	2.74	1.72 \times	2.93	1.74 \times	2.85

7. Mixed vs. All-Multimodal Training for HiViS

We conduct an additional experiment by replacing the text-only dataset in HiViS’s training with an equally size of samples’ multimodal dataset. As shown in Figure 7, the mixed-dataset HiViS (multimodal + text) outperforms the all-multimodal variant on most tasks, and maintains only small gaps on the remaining ones. This outcome reflects a property of HiViS: since the drafter in HiViS operates purely in the fused language space and never observes raw visual tokens, its performance is primarily governed by how well it models the target VLM’s next-token distribution rather than how well it processes visual sequences. Text-only dataset, which contain longer sequences and a much richer vocabulary, provides stronger supervision for learning long-range language dependencies, which in turn yields a drafter that is more robust across tasks.

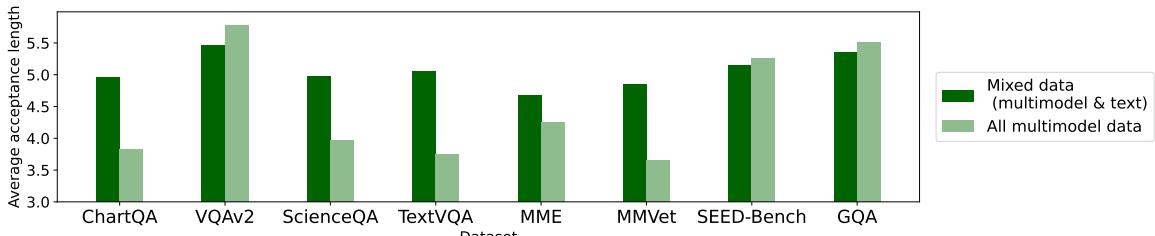


Figure 7. Comparison of average acceptance length between HiViS trained on mixed data (multimodal + text) and HiViS trained on multimodal-only data across several benchmarks, evaluated with a draft tree depth of 6 and temperature of 0.