HAWK: RETHINKING MULTIMODAL DRAFTING FOR SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding has achieved substantial lossless speedups for LLMs, but remains less effective for large vision-language models (LVLMs), where lightweight drafters struggle to use rich multimodal information. A second limitation is that standard distillation supervises the drafter only along the original training trajectory, without modeling how target predictions shift after the drafter's own proposals. As drafting moves away from this trajectory, the drafter can increasingly disagree with the target, reducing acceptance in later steps. We propose HAWK to address both limitations. HAWK uses representation similarity to select informative target layers and learns how to combine their hidden states. For visual information, it directly provides the drafter with compressed visual hidden states from the target model instead of raw visual tokens, making the visual information easier for a shallow drafter to use. HAWK also trains the drafter to capture how target predictions change after its own proposals, improving its agreement with the target during multi-step drafting. On SmolVLM-256M across ten multimodal benchmarks, HAWK raises average acceptance length from 3.32 to 4.08 and speedup from $2.19\times$ to $2.60\times$ over EAGLE-3 under greedy decoding, and from 2.89 to 3.41 and $1.92\times$ to $2.19\times$ under sampling.

1 INTRODUCTION

Large language models (LLMs) achieve strong performance across many tasks, but autoregressive generation remains expensive because each output token requires a sequential model forward pass (Ouyang et al., 2022; DeepSeek-AI, 2025; Leviathan et al., 2023). Speculative decoding reduces this cost by using a lightweight drafter to propose multiple tokens that are verified in parallel by the target model (Leviathan et al., 2023; Chen et al., 2023). Recent methods improve the trade-off between drafting cost and target agreement through auxiliary prediction heads (Cai et al., 2024), early-exit drafting (Elhoushi et al., 2024; Liu et al., 2024), or lightweight drafters conditioned on target hidden states (Li et al., 2024a;b; 2025).

Extending speculative decoding to large vision-language models (LVLMs) is more difficult because lightweight drafters must also use rich visual context. Existing methods therefore reduce the visual input through token compression, adaptive selection, or by reusing visual information already encoded by the target (Kang et al., 2026; Hu et al., 2026; Huang et al., 2025; Xie et al., 2026). However, reducing the number of visual tokens does not necessarily make their information easier for a shallow drafter to use, and aggressive compression may discard useful information. HAWK instead gives the drafter the visual hidden states the target computes during prefill, so the drafter starts from features the target has already built rather than from raw image tokens.

A second limitation lies in draft-model training. Standard distillation supervises predictions along the original response trajectory, while multi-step drafting conditions on the drafter's own proposals. Once a proposed token changes the context, the target model's subsequent prediction may also change. Existing approaches improve supervision at such states by matching target logits or features, including states induced by the drafter (Li et al., 2025; Zhang et al., 2025; Zhou et al., 2024; Lei et al., 2026). We instead supervise *how* the target prediction changes relative to the original trajectory. Since standard distillation already teaches the target distribution on the original context, this objective targets only what changes once the drafter's token enters the context.

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

Based on these observations, we propose HAWK, a multimodal speculative decoding framework that improves both the information provided to the drafter and the supervision used to train it. For target hidden states, rather than selecting layers only by depth, HAWK uses inter-layer representation similarity to identify distinct regions, selects representative layers within each region, and learns how to combine their features. For visual information, it reuses and compresses target visual hidden states computed during prefill, providing richer visual conditioning without requiring the drafter to recover the same information from raw visual tokens. During training, HAWK additionally substitutes the drafter's own proposals into the context and trains the drafter to match the resulting shift in target predictions.

Across ten multimodal benchmarks on SmolVLM-256M, HAWK consistently improves speculative drafting. Under greedy decoding, it increases average acceptance length from 3.32 to 4.08 and speedup ratio from $2.19\times$ to $2.60\times$ over EAGLE-3, while also outperforming ViSpec and HiViS on average acceptance length and speedup. Under sampling, it improves average acceptance length from 2.89 to 3.41 and speedup ratio from $1.92\times$ to $2.19\times$ over EAGLE-3. On the larger LLaVA-1.6-7B and Qwen2.5-VL-7B targets, HAWK raises average acceptance length under greedy decoding from 3.73 to 4.02 and from 3.19 to 3.40, and improves both acceptance length and speedup on every benchmark at both temperatures.

Our contributions are twofold:

- We improve multimodal drafting inputs through representation-based target-layer selection, learned feature fusion, and compressed visual hidden states reused from target prefill.
- We complement standard distillation with supervision on target prediction shifts caused by the drafter's own proposals, improving training for multi-step drafting without additional inference-time computation.

2 RELATED WORK

Speculative decoding for language models. Speculative decoding verifies several drafted tokens in one target forward pass (Leviathan et al., 2023; Chen et al., 2023). Later methods differ in where those drafts come from. Medusa attaches extra decoding heads to the target (Cai et al., 2024). LayerSkip and Kangaroo draft by exiting or reusing the target's own early layers, and LEAP searches over which layers to reuse (Elhoushi et al., 2024; Liu et al., 2024; Zheng & Liu, 2026). EAGLE instead trains a separate lightweight drafter that reads target hidden states, with later versions adding dynamic draft trees and features from several layers (Li et al., 2024a;b; 2025). On the training side, EAGLE-3's training-time test feeds the drafter its own features during training (Li et al., 2025), and HASS aligns training with the drafter's own contexts (Zhang et al., 2025); in both, the input tokens remain those of the training sequence. DistillSpec distills the target distribution on drafter-generated sequences (Zhou et al., 2024), and Draft-OPD applies on-policy distillation to speculative draft models (Lei et al., 2026). HAWK keeps the EAGLE-3 drafter and instead supervises how the target prediction changes after a draft token alters the context.

Speculative decoding for vision-language models. Vision-language speculative decoding must provide visual context to the drafter without incurring the full cost of the target model. Early approaches construct multimodal drafters using target vision encoders or specialized multimodal training (Ganesan et al., 2025; Lin et al., 2025). More recent methods increasingly reuse target-model information: DREAM injects intermediate target representations, while SpecVLM conditions an EAGLE-style drafter on target features together with adaptive visual compression (Hu et al., 2026; Huang et al., 2025). ViSpec learns compact visual representations for a shallow drafter, whereas HiViS avoids explicit visual-token processing by transferring target representations in which visual and textual information have already been fused (Kang et al., 2026; Xie et al., 2026). HAWK also builds on target representations, and differs in how it chooses and fuses target layers and in passing pooled target visual states directly to the drafter.

Layer similarity. Representation-similarity measures such as CKA (Kornblith et al., 2019) show that adjacent layers of deep networks are often highly similar, and layer-pruning work uses this redundancy to remove layers of LLMs (Men et al., 2024; Gromov et al., 2024). HAWK uses the same signal to choose which target layers a drafter reads.

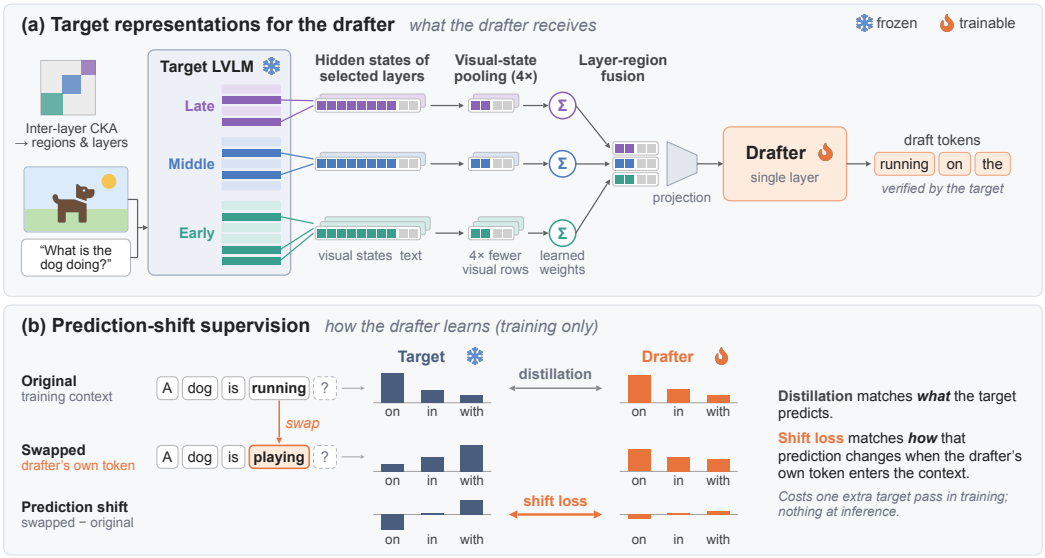


Figure 1: Overview of HAWK. (a) Target representations passed to the drafter. Inter-layer CKA divides the frozen target into early, middle, and late regions and selects representative layers. Visual hidden states from those layers are pooled by $4\times$, then combined within each region by learned weights. The three fused streams are projected into a single-layer drafter, whose proposals the target verifies. (b) Prediction-shift supervision, used only during training. Standard distillation matches the target's next-token prediction on the original context. The shift loss matches how that prediction changes after the drafter's own token replaces the original token.

3 METHOD

HAWK changes two parts of an EAGLE-3 drafter: the target information it receives (Sections 3.1 and 3.2) and how it is trained (Section 3.3). Figure 1 summarizes both.

3.1 LAYER-REGION FUSION

EAGLE-3 combines hidden states from three target layers at fixed depths, $(2, N/2, N-3)$ in 1-based indexing (Li et al., 2025). For SmolVLM-256M these are decoder blocks $[1, 14, 26]$ in the 0-based indexing we use (Appendix A). Fixed positions need not match where the target representations change, and reading more layers does not necessarily help, because nearby layers are often redundant. We therefore split the target into regions by representation similarity, select a few layers in each region, and learn how to combine them.

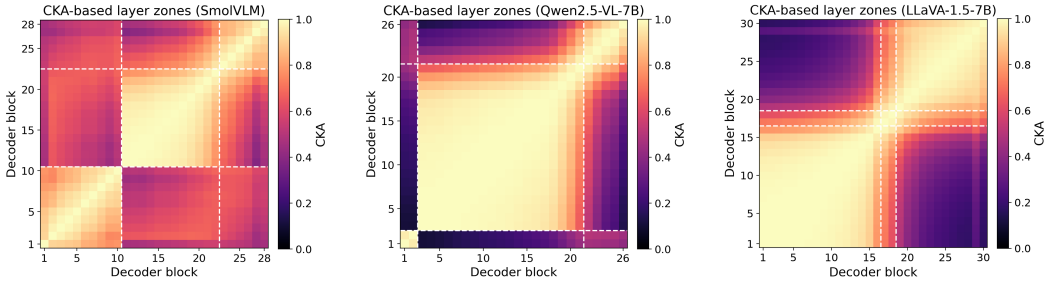
Region and layer selection. We measure layer similarity with linear CKA (Kornblith et al., 2019). For hidden states $X \in \mathbb{R}^{n \times d_X}$ and $Y \in \mathbb{R}^{n \times d_Y}$ of two layers at the same n token positions, with columns centered,

$$\mathrm{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F}. \tag{1}$$

We compute CKA on the text positions of 48 training conversations, after removing 32 outlier hidden dimensions (Appendix D). An exhaustive search over two boundaries then splits the target into three contiguous regions whose layers are more similar to each other than to layers in other regions. Within region $\mathcal{R}_r$, the CKA coverage of a subset $\mathcal{S}_r \subseteq \mathcal{R}_r$ is

$$C_{\mathrm{CKA}}(\mathcal{S}_r) = \frac{1}{|\mathcal{R}_r|} \sum_{l \in \mathcal{R}_r} \max_{s \in \mathcal{S}_r} \mathrm{CKA}(l, s). \tag{2}$$

Starting from an empty set, we add the layer with the largest coverage gain until the gain falls below $\epsilon_{\mathrm{CKA}} = 0.008$. Figure 2 shows the resulting regions; Appendix D gives the coverage curves and selected layers.



(a) SmolVLM-256M. (b) Qwen2.5-VL-7B. (c) LLaVA-1.6-7B.

Figure 2: Inter-layer CKA and the selected early, middle, and late regions. The boundaries differ across models, so one fixed depth rule does not fit all three.

Learnable fusion. Each region feeds one of the three EAGLE-3 input streams. We mix its selected layers with learned softmax weights,

$$\bar{\mathbf{H}}_r = \sum_{s \in \mathcal{S}_r} \alpha_{r,s} \tilde{\mathbf{H}}^{(s)}, \qquad \alpha_{r,s} = \frac{\exp(a_{r,s})}{\sum_{j \in \mathcal{S}_r} \exp(a_{r,j})}, \tag{3}$$

where $a_{r,s}$ are learnable scalars and $\tilde{\mathbf{H}}^{(s)}$ is the pooled hidden-state sequence of layer s (Section 3.2). We then apply the EAGLE-3 stream normalization $\mathcal{N}_r$ and input projection P,

$$\mathbf{U} = P\big([\mathcal{N}_{\text{early}}(\bar{\mathbf{H}}_{\text{early}}); \mathcal{N}_{\text{middle}}(\bar{\mathbf{H}}_{\text{middle}}); \mathcal{N}_{\text{late}}(\bar{\mathbf{H}}_{\text{late}})]\big), \tag{4}$$

so the drafter keeps the EAGLE-3 input interface. Figure 4 shows the learned weights.

3.2 Visual State Pooling

Recent multimodal drafters shorten the visual input with learned compression or drop visual tokens altogether (Kang et al., 2026; Xie et al., 2026). We instead give the drafter the visual hidden states that the target has already computed during prefill, and pool them to keep the sequence short. At each selected layer l, we split every contiguous run of image rows (one 64-row tile in SmolVLM) into consecutive groups $\mathcal{G}_j$ of c rows and average each group:

$$\tilde{\mathbf{h}}_j^{(l)} = \frac{1}{|\mathcal{G}_j|} \sum_{i \in \mathcal{G}_j} \mathbf{h}_i^{(l)}. \tag{5}$$

All layers use the same groups, text rows are left unchanged, and each pooled row keeps the position id of the first row in its group (Appendix A). We use $c = 4$. Moderate pooling improves acceptance, while stronger pooling removes useful information (Section 4.3, Appendix F). Pooling changes only the drafter's input; the target still processes the full visual sequence during prefill and verification.

3.3 Prediction-Shift Supervision

Standard distillation trains the drafter to match target predictions along the training sequence. In multi-step drafting, however, the drafter conditions on its own proposals, and even a plausible alternative token can change what the target predicts next. We therefore train the drafter so that the change in its prediction after such a substitution matches the change in the target's prediction.

Alternative token selection. Let x_t be the response token at position t. At the first rollout step of EAGLE-3's training-time test (one unrolled draft step), the drafter proposes $\hat{x}_t = \arg\max_x p_D(x \mid x_{<t})$. We keep the positions where this proposal differs from the target's top-1 prediction:

$$\mathcal{S} = \big\{t : \hat{x}_t \neq \arg\max_x p_T(x \mid x_{<t})\big\}. \tag{6}$$

For each $t \in \mathcal{S}$, the drafter runs its second rollout step with $\hat{x}_t$ in place of x_t. Its own output feature from the first step and the prefix cache stay the same; only the input token changes. EAGLE-3's training-time test trains the drafter on its own feature drift but never on a context whose token

differs from the training sequence; the substitution supervises exactly this token pathway, at the same rollout state (Appendix G).

Logit-shift objective. Let $z^{\text{orig}}_{T,t}$ and $z^{\text{orig}}_{D,t}$ be the target and drafter logits for the token following position t under the original context, and $z^{\text{alt}}_{T,t}$ and $z^{\text{alt}}_{D,t}$ the logits after substitution. We map the target logits to the draft vocabulary V_D and center every logit vector, $\text{center}(z) = z - \text{mean}(z)$, which removes offsets that softmax ignores. The target and drafter shifts are

$$\Delta_{T,t} = \text{center}(z^{\text{alt}}_{T,t}) - \text{center}(z^{\text{orig}}_{T,t}), \qquad \Delta_{D,t} = \text{center}(z^{\text{alt}}_{D,t}) - \text{center}(z^{\text{orig}}_{D,t}), \tag{7}$$

and the training objective is

$$\mathcal{L} = \mathcal{L}_{\text{std}} + \lambda_{\text{shift}}\mathcal{L}_{\text{shift}}, \qquad \mathcal{L}_{\text{shift}} = \underset{t\in\mathcal{S}}{\text{mean}} \frac{1}{|V_D|}\left\|\Delta_{D,t} - \Delta_{T,t}\right\|_2^2, \qquad \lambda_{\text{shift}} = 0.1, \tag{8}$$

where $\mathcal{L}_{\text{std}}$ is the standard distillation loss. Because centered logits equal log-probabilities up to a shared constant, $\Delta_{T,t}$ is the centered log-ratio $\log p^{\text{alt}}_T - \log p^{\text{orig}}_T$. $\mathcal{L}_{\text{std}}$ fits the level of the prediction, and $\mathcal{L}_{\text{shift}}$ fits how it changes (Appendix G). As a function of the input token, $\Delta_{T,t}$ is a finite difference of the target's log-probabilities along $x_t \to \hat{x}_t$, so $\mathcal{L}_{\text{shift}}$ matches the drafter's sensitivity to its input token to that of the target.

Batched target supervision. Scoring each substitution separately would need one target forward pass per selected position. We instead substitute all positions in $\mathcal{S}$ into one copy of the sequence and obtain z^{alt}_T with a single extra target pass. The original logits z^{orig}_T and the target hidden states come from the offline cache. The extra pass runs only during training and adds no inference cost.

4 Experiments

4.1 Experimental Setup

Models and benchmarks. We use SmolVLM-256M-Instruct (Marafioti et al., 2025) as the main target and report LLaVA-1.6-7B and Qwen2.5-VL-7B in Section 4.4. We evaluate on ten benchmarks: MMMU, SEED-Bench, MME, VQAv2, and MM-Vet for general understanding; MathVista, ScienceQA, and GQA for visual reasoning; and TextVQA and DocVQA for OCR (Yue et al., 2024; Li et al., 2023; Fu et al., 2026; Goyal et al., 2017; Yu et al., 2023; Lu et al., 2024; 2022; Hudson & Manning, 2019; Singh et al., 2019; Mathew et al., 2021). Each benchmark uses 80 prompts (Appendix H).

Baselines. We compare with EAGLE-3 (Li et al., 2025), trained with the same data, offline cache, and budget as HAWK, and with ViSpec (Kang et al., 2026) and HiViS (Xie et al., 2026), trained with their official implementations. All methods use the same evaluator, draft-tree configuration, and prompts (Appendix C).

Metrics. Following HiViS, we report the speedup ratio (SR), the generation token rate relative to autoregressive decoding, and the average acceptance length τ, the number of tokens produced per verification step (Appendix B). We do not report task accuracy, since speculative decoding leaves the target's output distribution unchanged.

Training setup. Following HiViS, we train on 68K text-only conversations from ShareGPT and 68K multimodal conversations from LLaVA-Mix665K (ShareGPT, 2023; Liu et al., 2023), with responses regenerated by the target. The EAGLE-3 and HAWK drafters have a single decoder layer and are trained for 20 epochs (Appendix A).

Inference setup. Following ViSpec and HiViS, we evaluate greedy decoding ($T = 0$) and sampling ($T = 1$). All runs use one NVIDIA RTX A5000 GPU with batch size 1. The draft tree has depth 5, top-10 expansion, and 60 candidate tokens (Appendix B).

4.2 Main Results

Table 1 compares HAWK with EAGLE-3, ViSpec, and HiViS on SmolVLM-256M. HAWK has the highest average speedup ratio and acceptance length at both temperatures.

Table 1: Speedup ratio (SR) and average acceptance length (τ) on SmolVLM-256M under greedy decoding ($T = 0$, top) and sampling ($T = 1$, bottom). Averages at both temperatures cover ten benchmarks.

Model	Methods	General										Reasoning						OCR					
		MMMU		SEED-Bench		MME		VQAv2		MM-Vet		MathVista		SQA		GQA		TextVQA		DocVQA		Avg	
		SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ
Greedy decoding ($T = 0$)																							
SmolVLM 256M	EAGLE-3	1.99	2.87	2.42	4.29	2.27	3.87	2.45	4.16	2.15	3.40	2.44	2.99	1.94	2.75	2.54	4.28	2.11	2.56	1.61	2.00	2.19	3.32
	ViSpec	1.62	2.24	2.34	4.38	2.06	3.56	2.35	4.48	2.04	3.47	2.38	3.32	1.84	2.74	2.34	4.16	2.27	3.02	1.57	2.00	2.08	3.34
	HiViS	1.98	2.70	2.60	4.73	2.37	4.01	2.72	4.71	2.40	3.85	3.00	3.87	2.24	3.25	2.77	4.70	2.56	3.25	1.77	2.10	2.44	3.72
	HAWK	**2.23**	**3.39**	**2.74**	**5.02**	**2.55**	**4.54**	**2.73**	**4.80**	**2.42**	**4.04**	**3.24**	**4.26**	**2.35**	**3.54**	**2.82**	**4.90**	**2.85**	**3.63**	**2.03**	**2.71**	**2.60**	**4.08**
Sampling ($T = 1$)																							
Model	Methods	General										Reasoning						OCR					
		MMMU		SEED-Bench		MME		VQAv2		MM-Vet		MathVista		SQA		GQA		TextVQA		DocVQA		Avg	
		SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ
SmolVLM 256M	EAGLE-3	1.65	2.42	2.26	3.88	2.03	3.28	2.20	3.67	1.66	2.64	2.04	2.79	1.68	2.35	2.15	3.48	2.05	2.48	1.45	1.93	1.92	2.89
	ViSpec	1.42	1.99	2.19	4.00	1.85	2.99	2.22	**4.17**	1.75	2.83	1.98	3.05	1.71	2.53	2.05	3.63	2.09	2.88	1.42	1.95	1.87	3.00
	HiViS	1.65	2.35	**2.47**	**4.38**	2.01	3.34	2.32	3.76	**1.92**	**3.13**	2.37	3.37	1.90	2.82	2.35	3.95	2.35	3.10	1.58	2.06	2.09	3.23
	HAWK	**1.72**	**2.68**	2.43	4.27	**2.14**	**3.48**	**2.44**	4.09	1.91	3.03	**2.51**	**3.73**	**1.96**	**3.01**	**2.39**	**3.99**	**2.63**	**3.34**	**1.77**	**2.46**	**2.19**	**3.41**

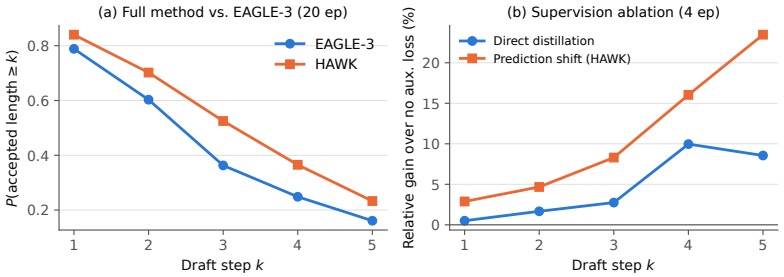


Figure 3: (a) Probability that the first k draft tokens are all accepted, EAGLE-3 vs. HAWK. (b) Relative change of the same quantity over the drafter trained without auxiliary loss, for the supervision variants of Table 3. The gain from prediction-shift supervision grows from 3% at $k = 1$ to 23% at $k = 5$, against at most 10% for direct distillation.

Greedy decoding. At $T = 0$, HAWK has the highest SR and τ on all ten benchmarks. Its average τ is 4.08, compared with 3.32 for EAGLE-3, 3.34 for ViSpec, and 3.72 for HiViS, and its average SR is 2.60×, compared with 2.19×, 2.08×, and 2.44×. The largest speedup is 3.24× on MathVista, where τ rises from 2.99 with EAGLE-3 to 4.26.

Sampling. At $T = 1$, HAWK again has the best averages. Compared with EAGLE-3, it raises τ from 2.89 to 3.41 and SR from 1.92× to 2.19×, with gains on all ten benchmarks. It also exceeds ViSpec (3.00, 1.87×) and HiViS (3.23, 2.09×) on average.

Gains by task type. On SmolVLM-256M under greedy decoding, τ increases by 14–19% on the five general benchmarks and GQA, and by 29–42% on MathVista, ScienceQA, TextVQA, and DocVQA. The margin over HiViS is largest on MMMU (3.39 vs. 2.70) and DocVQA (2.71 vs. 2.10), where HiViS gains little or nothing over EAGLE-3. The largest relative gains are on reasoning and OCR benchmarks, whose answers depend most on visual detail.

Acceptance by draft step. The gain over EAGLE-3 also grows with draft depth (Figure 3a). The probability that the first k draft tokens are all accepted rises from 0.789 to 0.840 at $k = 1$ and from 0.161 to 0.233 at $k = 5$, a relative gain of 6% at the first step and 45% at the fifth. Isolating prediction-shift supervision in the four-epoch ablation gives the same pattern: relative to the drafter trained without it, the gain grows from 3% at $k = 1$ to 23% at $k = 5$, against at most 10% for direct distillation on the substituted contexts (Figure 3b). Later steps are where the drafter conditions on its own proposals, which is the case prediction-shift supervision is designed for.

4.3 Ablation Studies

We ablate the main design choices of HAWK on SmolVLM-256M. All variants are trained for four epochs and evaluated under greedy decoding ($T = 0$) on six benchmarks: SEED-Bench, MM-Vet, MathVista, ScienceQA, GQA, and DocVQA. We report acceptance length τ and speedup ratio (SR). Each study isolates one design choice while disabling the other HAWK components.

Visual-state pooling. Without pooling, the drafter receives all visual hidden-state rows (832 for a 13-tile image) and achieves an average acceptance length of 3.05 with a $2.07\times$ speedup. As shown in Table 2, $4\times$ pooling reduces the sequence to 208 rows while improving both metrics to $\tau = 3.10$ and $2.10\times$. More aggressive $16\times$ pooling leaves only 52 rows and reduces performance to $\tau = 3.04$ and $2.07\times$, roughly returning to the no-pooling baseline. We therefore use $4\times$ pooling, which cuts the visual sequence by 75% while improving both metrics.

Table 2: Ablation of visual-state pooling on SmolVLM-256M. The other components are disabled.

Setting	SEED-Bench	MM-Vet	MathVista	SQA	GQA	DocVQA	Avg.
Acceptance length τ							
No pooling	4.03	3.15	2.85	2.36	3.80	2.08	3.05
$4\times$ pooling	4.03	3.25	2.92	2.45	3.86	2.09	**3.10**
$16\times$ pooling	4.00	3.20	2.75	2.43	3.82	2.03	3.04
Speedup ratio							
No pooling	2.36	2.04	2.18	1.74	2.28	1.79	2.07
$4\times$ pooling	2.39	2.10	2.21	1.81	2.31	1.77	**2.10**
$16\times$ pooling	2.38	2.07	2.18	1.81	2.29	1.67	2.07

Prediction-shift supervision. For this study, we use the original target layers $[1, 14, 26]$ and disable visual-state pooling. Without the additional supervision, standard draft distillation gives $\tau = 3.05$ and a $2.07\times$ speedup. Applying prediction-shift supervision with a larger weight of 0.3 gives a similar result, reaching $\tau = 3.11$ and $2.05\times$ speedup. In contrast, HAWK's prediction-shift supervision improves both metrics, reaching $\tau = 3.17$ and $2.09\times$ speedup (Table 3).

Direct distillation (Table 3) uses the same substituted contexts, token selection, loss weight, and extra target pass, and differs only in the loss: cross-entropy to p_T^{alt} instead of squared error on the shift. It raises τ by 0.05 against 0.12 for the shift loss, and its speedup drops to $2.05\times$.

Table 3: Ablation of prediction-shift supervision on SmolVLM-256M. The other components are disabled. Direct distillation matches the target distribution on substituted contexts, while HAWK supervises the induced prediction shift.

Setting	SEED-Bench	MM-Vet	MathVista	SQA	GQA	DocVQA	Avg.
Acceptance length τ							
No auxiliary loss	4.03	3.15	2.85	2.36	3.80	2.08	3.05
Direct distillation	4.00	3.21	2.84	2.56	3.85	2.14	3.10
Shift, w=0.3	3.94	3.17	2.98	2.60	3.79	2.16	3.11
HAWK	4.01	3.29	2.97	2.68	3.88	2.16	**3.17**
Speedup ratio							
No auxiliary loss	2.36	2.04	2.18	1.74	2.28	1.79	2.07
Direct distillation	2.32	2.02	2.19	1.80	2.25	1.69	2.05
Shift, w=0.3	2.31	2.02	2.20	1.80	2.23	1.76	2.05
HAWK	2.33	2.06	2.22	1.88	2.27	1.78	**2.09**

Layer-region fusion. Finally, we separate the effects of representation-based layer selection and learned fusion. Visual-state pooling and prediction-shift supervision are disabled in this study. As a reference, the EAGLE-3 layers $[1, 14, 26]$ under the same conditions give $\tau = 3.05$ and $2.07\times$ (the no-pooling row of Table 2). Using uniformly spaced target layers with learned mixing gives an average acceptance length of 3.17 and a $2.09\times$ speedup. Replacing the uniform layer grid with CKA-selected layers while fixing the mixing weights to uniform averaging raises acceptance length to 3.21, with the same $2.09\times$ speedup. This comparison isolates the benefit of selecting layers according to representation similarity rather than position alone. Learning the mixing weights over

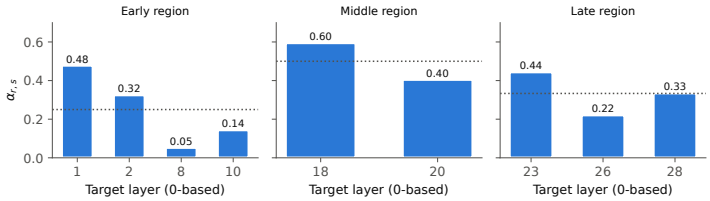


Figure 4: Learned fusion weights $\alpha_{r,s}$ of the HAWK drafter on SmolVLM-256M. Dotted lines mark uniform weights.

the same CKA-selected layers further improves acceptance length to 3.24 and speedup to 2.11× (Table 4).

Table 4: Ablation of layer selection and fusion on SmolVLM-256M. The other components are disabled. Uniform layers are evenly spaced. CKA layers are selected by representation similarity. Learned mixing trains the region weights; uniform mixing freezes them at equal weights.

Setting	SEED-Bench	MM-Vet	MathVista	SQA	GQA	DocVQA	Avg.
Acceptance length τ							
EAGLE-3 layers (reference)	4.03	3.15	2.85	2.36	3.80	2.08	3.05
Uniform layers, learned mix	4.16	3.25	2.86	2.65	3.92	2.19	3.17
CKA layers, uniform mix	4.07	3.29	3.09	2.66	3.93	2.21	3.21
CKA layers, learned mix	4.18	3.34	3.14	2.67	3.90	2.22	**3.24**
Speedup ratio							
EAGLE-3 layers (reference)	2.36	2.04	2.18	1.74	2.28	1.79	2.07
Uniform layers, learned mix	2.38	2.06	2.21	1.85	2.28	1.75	2.09
CKA layers, uniform mix	2.33	2.07	2.27	1.85	2.26	1.78	2.09
CKA layers, learned mix	2.38	2.05	2.32	1.84	2.26	1.80	**2.11**

4.4 LARGER TARGET MODELS

Table 5 reports results on LLaVA-1.6-7B and Qwen2.5-VL-7B. Due to limited resources, both EAGLE-3 and HAWK are trained on 30K examples for 10 epochs at this scale, and we do not retrain ViSpec or HiViS. Under greedy decoding, HAWK raises average acceptance length from 3.73 to 4.02 on LLaVA-1.6-7B and from 3.19 to 3.40 on Qwen2.5-VL-7B, and average speedup from 2.37× to 2.47× and from 2.17× to 2.25×. Both τ and SR improve on every benchmark at both temperatures.

Table 5: Speedup ratio (SR) and average acceptance length (τ) on LLaVA-1.6-7B and Qwen2.5-VL-7B under greedy decoding ($T = 0$, top) and sampling ($T = 1$, bottom). Both drafters are trained on 30K examples for 10 epochs.

Model	Methods	General										Reasoning						OCR				Avg	
		MMMU		SEED-Bench		MME		VQAv2		MM-Vet		MathVista		SQA		GQA		TextVQA		DocVQA			
		SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ
Greedy decoding ($T = 0$)																							
LLaVA-1.6 7B	EAGLE-3	2.32	3.59	2.57	4.30	2.29	3.51	2.66	4.53	2.33	3.62	2.34	3.65	2.18	3.20	2.58	4.31	2.28	3.49	2.15	3.13	2.37	3.73
	HAWK	2.41	3.84	2.73	4.74	2.37	3.73	2.79	4.92	2.43	3.89	2.43	3.90	2.28	3.49	2.68	4.60	2.39	3.79	2.21	3.28	2.47	4.02
Qwen2.5-VL 7B	EAGLE-3	1.99	2.67	2.39	3.79	2.11	3.00	2.48	4.05	2.16	3.15	2.10	2.99	2.07	2.89	2.41	3.85	2.06	2.87	1.96	2.60	2.17	3.19
	HAWK	2.10	2.98	2.51	4.12	2.17	3.18	2.62	4.43	2.22	3.32	2.16	3.15	2.12	3.03	2.49	4.07	2.11	3.00	2.00	2.71	2.25	3.40
Sampling ($T = 1$)																							
LLaVA-1.6 7B	EAGLE-3	1.83	2.68	2.19	3.55	1.90	2.86	2.26	3.71	1.95	2.96	1.86	2.76	1.83	2.69	2.19	3.54	1.81	2.64	1.74	2.45	1.96	2.98
	HAWK	1.92	2.90	2.23	3.64	2.00	3.08	2.38	4.02	2.02	3.15	1.93	2.92	1.93	2.93	2.28	3.77	1.88	2.79	1.81	2.63	2.04	3.18
Qwen2.5-VL 7B	EAGLE-3	1.74	2.47	2.16	3.47	1.87	2.78	2.18	3.52	1.93	2.91	1.85	2.72	1.86	2.74	2.07	3.26	1.75	2.48	1.68	2.31	1.91	2.87
	HAWK	1.82	2.65	2.23	3.64	1.92	2.90	2.27	3.74	1.96	2.99	1.94	2.95	1.89	2.82	2.16	3.49	1.78	2.57	1.72	2.41	1.97	3.02

5 CONCLUSION

We introduced HAWK to improve multimodal speculative decoding through better target conditioning and supervision. Layer-region fusion combines features from representation-based target regions, and visual-state pooling compresses target-processed visual states into a shorter sequence. Prediction-shift supervision teaches the drafter to match target logit shifts after plausible token substitutions. On SmolVLM-256M, HAWK improves acceptance length and speedup over EAGLE-3, ViSpec, and HiViS under both greedy decoding and sampling, and on LLaVA-1.6-7B and Qwen2.5-VL-7B it improves both metrics over EAGLE-3 on every benchmark at both temperatures.

REPRODUCIBILITY STATEMENT

Appendix A lists the drafter architecture, optimizer, learning-rate schedule, training data, the target layers read by every reported drafter, and the prediction-shift hyperparameters. Appendix B gives the hardware, decoding configuration, and metric definitions, and Appendix H the source, split, and prompt selection of every benchmark. Appendix D gives the CKA data, the region objective, and the layer-selection algorithm with its selected layers. Appendix C describes how the baselines were run. We will release code for training, CKA layer selection, and evaluation, together with the trained drafters.

AI USE STATEMENT

We used generative AI tools solely to improve the manuscript's writing, including grammar, phrasing, and clarity. All AI-assisted text was carefully reviewed and edited by the authors. Generative AI was not used to develop research ideas, formulate theoretical claims, or produce experimental results.

REFERENCES

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.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12622–12642, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Advances in Neural Information Processing Systems*, 38, 2026.

Mugilan Ganesan, Shane Segal, Ankur Aggarwal, Nish Sinnadurai, Sean Lie, and Vithursan Thangarasa. Massv: Multimodal adaptation and self-data distillation for speculative decoding of vision-language models. In *EMNLP (Findings)*, pp. 12265–12276, 2025.

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*, pp. 6904–6913, 2017.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

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. *Advances in Neural Information Processing Systems*, 38:167592–167612, 2026.

Haiduo Huang, Fuwei Yang, Zhenhua Liu, Xuanwu Yin, Dong Li, Pengju Ren, and Emad Barsoum. Specvlm: Fast speculative decoding in vision-language models. *arXiv preprint arXiv:2509.11815*, 2025.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Jialiang Kang, Han Shu, Wenshuo Li, Yingjie Zhai, and Xinghao Chen. Vispec: Accelerating vision-language models with vision-aware speculative decoding. *Advances in Neural Information Processing Systems*, 38:115511–115532, 2026.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3519–3529, 2019.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, Xiaoye Qu, Ganqu Cui, Bowen Zhou, Ning Ding, Yun Luo, et al. Draft-opd: On-policy distillation for speculative draft models. *arXiv preprint arXiv:2605.29343*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

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.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning*, 2024a.

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*, 2024b.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *Annual Conference on Neural Information Processing Systems*, 2025.

Luxi Lin, Zhihang Lin, Zhanpeng Zeng, and Rongrong Ji. Speculative decoding reimagined for multimodal large language models. *arXiv preprint arXiv:2505.14260*, 2025.

Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Kai Han, and Yunhe Wang. Kangaroo: Lossless self-speculative decoding via double early exiting. *arXiv preprint arXiv:2404.18911*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Pan Lu, Swaroop Mishra, Tony 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. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *WACV*, 2021.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL `https://arxiv.org/abs/2203.02155`.

ShareGPT. ShareGPT. `https://sharegpt.com/`, 2023.

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*, pp. 8317–8326, 2019.

Zhinan Xie, Peisong Wang, Shuang Qiu, and Jian Cheng. Hivis: Hiding visual tokens from the drafter for speculative decoding in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8952–8961, 2026.

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, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. Learning harmonized representations for speculative sampling. In *International Conference on Learning Representations*, 2025.

LeiQuan Zheng and Yuan Liu. Leap: Zone-aware mcts for llm self-speculative decoding. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, 2024.

A Implementation Details

Layer indexing. Target layers are indexed from 0 by decoder block: layer l is the output of the l-th decoder block, i.e., `hidden_states[`l+1`]` in HuggingFace convention, where `hidden_states[0]` is the embedding output. SmolVLM-256M has $N = 30$ decoder blocks, Qwen2.5-VL-7B has $N = 28$, and LLaVA-1.6-7B has $N = 32$. In this indexing the EAGLE-3 rule selects blocks $(1,\ N/2 - 1,\ N - 4)$, e.g., $[1, 14, 26]$ for SmolVLM-256M; this is $(2,\ N/2,\ N - 3)$ in the 1-based notation of Li et al. (2025). The CKA search ranges over blocks 1 to $N - 2$.

Drafter. A single-layer EAGLE-3 drafter (hidden size 576, 9 attention heads, 3 KV heads, intermediate size 1536) with a 32,000-token draft vocabulary mapped from the 49,280-token target vocabulary, in bfloat16. Training-time test unrolls 7 draft steps.

Optimization. AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay), constant learning rate 10^{-4} without warmup, gradient clipping at 1.0, bfloat16, seed 42, 20 epochs, on 8 NVIDIA RTX A5000 GPUs.

Training data. Following HiViS (Xie et al., 2026), we combine 68,000 text-only conversations from ShareGPT with 68,000 multimodal conversations from LLaVA-Mix665K, filter out examples whose responses contain fewer than five words, and regenerate assistant responses with the target model, giving 132,929 training records. Target hidden states of the selected layers and target logits are pre-computed once and stored as an offline cache.

Visual-state pooling. SmolVLM encodes an image as 9, 13, or 17 tiles depending on aspect ratio, with 64 image rows per tile (576, 832, or 1,088 rows; 832 for the 13-tile case quoted in the main text). Pooling acts on each contiguous run of image rows independently. With $c = 4$ each tile keeps 16 rows (208 for 13 tiles); the $16\times$ setting keeps 4 rows per tile (52 for 13 tiles). Each pooled row takes the token id and position id of the first row in its group, so text rows keep their original absolute positions and the RoPE positions seen by the drafter are a subset of those seen by the target. Within a tile, rows follow raster order over an 8×8 grid, so a group of four consecutive rows covers a 1×4 horizontal strip of patches. The EAGLE-3 baseline receives all image rows.

Target layers. Table 6 lists the target layers read by each SmolVLM-256M drafter.

Table 6: Target layers (0-based decoder blocks) read by each SmolVLM-256M drafter.

Drafter	Early / Middle / Late	Used in
EAGLE-3	[1] / [14] / [26]	Table 1; Tables 2–3
HAWK	[1, 2, 8, 10] / [18, 20] / [23, 26, 28]	Table 1
CKA coverage, $\epsilon_{\mathrm{CKA}} = 0.008$	[1, 2, 5, 9] / [14, 18, 21] / [23, 25, 27]	Table 4
Uniform grid	[1, 4, 8, 11] / [14, 18] / [21, 24, 28]	Table 4

Prediction-shift supervision. $\lambda_{\mathrm{shift}} = 0.1$, drafter proposal = drafter top-1, substitution at the first rollout step only.

B Hardware and Timing Protocol

All evaluation runs on a single NVIDIA RTX A5000 (24 GB) with the other GPUs on the machine idle, one prompt at a time (batch size 1), in bfloat16, with three warmup prompts before timing. Tree decoding uses depth 5, 60 candidate tokens, and top-k 10. Prompts are limited to 4,000 input tokens and generation to 500 new tokens. At $T = 1$, the target samples from its full output distribution without top-k or nucleus truncation. Acceptance length τ is, per prompt, the mean over verification rounds of the number of accepted draft tokens plus one, averaged over prompts. Speedup ratio follows the definition of HiViS (Xie et al., 2026): per prompt, the method's generation token rate divided by that of autoregressive decoding, averaged over prompts. Greedy results are reported on prompts whose decoded answer is identical to the autoregressive answer for every method in the table.

C BASELINE REPRODUCTION

EAGLE-3 uses the same target, training data, offline cache, optimizer, and number of epochs as HAWK, reads the EAGLE-3 layers in Table 6, and receives all image rows. ViSpec and HiViS are trained with their official implementations. All four methods are evaluated with the same evaluator, decoding loop, tree configuration, prompts, and prompt filter, each against an autoregressive baseline run on the same machine.

D CKA LAYER SELECTION

Data. We run each target on 48 training conversations and pool representations over all text positions (6,438 for SmolVLM-256M, 11,193 for Qwen2.5-VL-7B, 10,177 for LLaVA-1.6-7B). Image positions are excluded: the drafting loss is applied at text positions, and hundreds of near-duplicate visual rows would otherwise dominate the statistic.

Outlier dimensions. From block 11 onward in SmolVLM-256M, a single hidden dimension carries most of the variance, and with it included linear CKA between any two adjacent layers in blocks 11–28 is 1.000. We therefore remove the 32 highest-variance dimensions, chosen once over all layers so that every layer is compared in the same subspace, before computing CKA.

Region boundaries. Let $\mathcal{L} = \{1, \dots, N-2\}$. We search over all partitions of $\mathcal{L}$ into three contiguous regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ of at least two layers each and choose

$$(\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3) = \arg\max \underset{\substack{r;\, i<j \\ i,j \in \mathcal{R}_r}}{\operatorname{mean}} \mathrm{CKA}(i,j) \; - \underset{\substack{r<r' \\ i \in \mathcal{R}_r,\, j \in \mathcal{R}_{r'}}}{\operatorname{mean}} \mathrm{CKA}(i,j), \tag{9}$$

by exhaustive search over the $O(N^2)$ boundary pairs.

Layer selection. Within each region, Algorithm 1 greedily adds the layer that most increases C_{CKA} until the gain falls to $\epsilon_{\mathrm{CKA}} = 0.008$ or below. The first layer added is the region's coverage medoid. Table 7 lists the regions and selected layers of each target, and Figure 5 shows how coverage grows as layers are added.

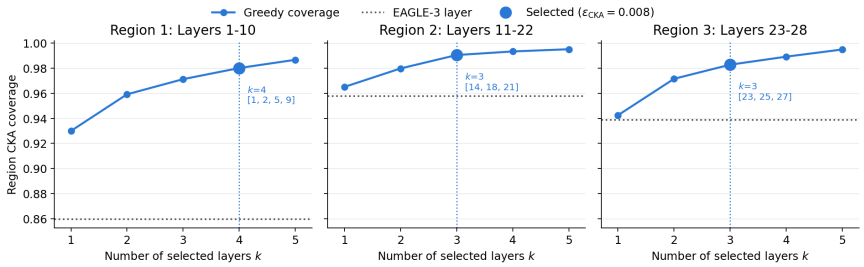
Algorithm 1 Greedy CKA-coverage selection for region $\mathcal{R}$

$\mathcal{S} \leftarrow \emptyset$
while $\mathcal{R} \setminus \mathcal{S} \neq \emptyset$:
  $s^\star \leftarrow \arg\max_{s \in \mathcal{R} \setminus \mathcal{S}} C_{\mathrm{CKA}}(\mathcal{S} \cup \{s\})$
  $\delta \leftarrow C_{\mathrm{CKA}}(\mathcal{S} \cup \{s^\star\}) - C_{\mathrm{CKA}}(\mathcal{S})$
  if $\mathcal{S} \neq \emptyset$ **and** $\delta \leq \epsilon_{\mathrm{CKA}}$: **break**
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{s^\star\}$
return $\mathcal{S}$

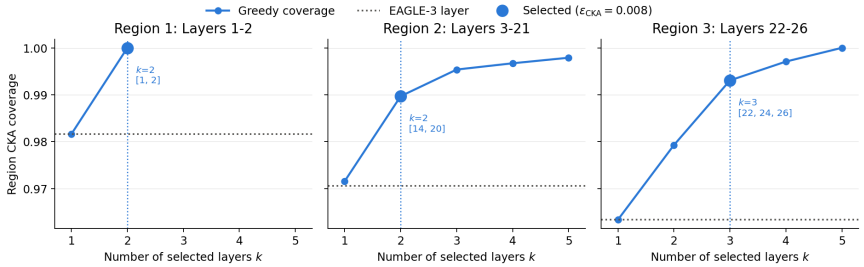
Table 7: CKA regions and coverage selections (0-based blocks, $\epsilon_{\mathrm{CKA}} = 0.008$). Layers are listed in order of addition; parentheses give each layer's coverage gain.

Target	Early	Middle	Late
SmolVLM-256M	1–10: 5, 9, 2, 1 (.930, .029, .012, .009)	11–22: 18, 21, 14 (.965, .015, .011)	23–28: 25, 27, 23 (.943, .029, .011)
Qwen2.5-VL-7B	1–2: 1, 2 (.982, .018)	3–21: 14, 20 (.972, .018)	22–26: 24, 26, 22 (.963, .016, .014)
LLaVA-1.6-7B	1–16: 10, 15 (.980, .011)	17–18: 17, 18 (.990, .010)	19–30: 25, 20, 28 (.964, .016, .009)

Qwen2.5-VL-7B Region 1. The early region of Qwen2.5-VL-7B contains only blocks 1–2 because the transition from block 2 to block 3 is the sharpest in the network: CKA between blocks 2 and 3 is 0.10, while every adjacent pair from block 3 to block 26 has CKA of at least 0.95. The mean CKA between blocks 1–2 and blocks 3–21 is 0.22 (Figure 2b).



(a) SmolVLM-256M.



(b) Qwen2.5-VL-7B.

Figure 5: Region-wise CKA coverage as the number of selected target layers increases for (a) SmolVLM-256M and (b) Qwen2.5-VL-7B. Large markers show where the greedy selection stops at $\epsilon_{\text{CKA}} = 0.008$; dotted lines show the coverage of the single EAGLE-3 layer in each region.

E Learned Fusion Weights

Figure 4 shows the softmax-normalized fusion weights $\alpha_{r,s}$ of the HAWK drafter after training. The weights depart from uniform in every region: blocks 1 and 2 receive 0.48 and 0.32 of the early region while block 8 receives 0.05; block 18 receives 0.60 of the middle region; and block 23 receives 0.44 of the late region.

F Visual Compression Diagnostic

Figure 6 relates the compression factor of the visual hidden states to drafting and to the target's own answers. Panel (a) reports the acceptance length of drafters conditioned on pooled target visual states: $4\times$ pooling gives the highest acceptance length, and $16\times$ pooling falls back toward the uncompressed setting. Panel (b) is a separate target-only experiment in which the target itself receives the compressed visual input; its accuracy drops from 49.1 to 39.0 at $4\times$ and to 29.3 at $16\times$. Panel (b) only measures how much information compression removes. In every speculative-decoding result in this paper the target receives the full visual input, so compression changes what the drafter sees but never the target's output.

G Prediction-Shift Supervision: Details

The shift is a centered log-probability ratio. For logits z and $p = \text{softmax}(z)$, $\log p_i = z_i - \log \sum_j e^{z_j}$, so $\text{center}(z)_i = \log p_i - \frac{1}{|V|}\sum_j \log p_j$. Hence

$$\Delta_{T,t,i} = \log \frac{p_T^{\text{alt}}(i)}{p_T^{\text{orig}}(i)} - \frac{1}{|V|}\sum_j \log \frac{p_T^{\text{alt}}(j)}{p_T^{\text{orig}}(j)}, \tag{10}$$

and likewise for $\Delta_{D,t}$. $\mathcal{L}_{\text{shift}} = 0$ exactly when $p_D^{\text{alt}}(i)/p_D^{\text{orig}}(i) \propto p_T^{\text{alt}}(i)/p_T^{\text{orig}}(i)$ for every token i. Combined with $\mathcal{L}_{\text{std}}$, which fits p_D^{orig} to p_T^{orig}, this fits p_D^{alt} to p_T^{alt} through the ratio rather than through the level: $\mathcal{L}_{\text{std}}$ supervises the prediction, and $\mathcal{L}_{\text{shift}}$ supervises how the prediction changes when the

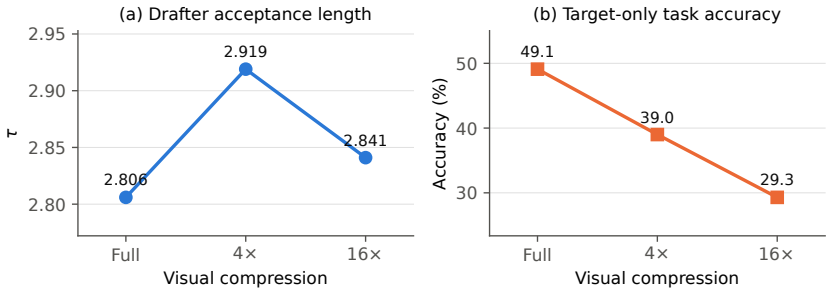


Figure 6: Visual compression diagnostic. (a) Acceptance length of drafters conditioned on pooled target visual states. (b) Task accuracy of the target when its own visual input is compressed by the same factor.

input token changes. As a function of the input token, Δ is a finite difference of the log-probabilities along the direction $x_t \to \hat{x}_t$. Unlike cross-entropy on p_T^{alt}, the squared error on centered logits weights every draft-vocabulary entry equally rather than concentrating on high-probability tokens. The squared error is averaged over the draft vocabulary.

Relation to training-time test. In EAGLE-3's training-time test, the drafter at rollout step $k > 1$ receives its own output feature from step $k-1$, but its input token is always the training-sequence token x_t. It is therefore trained on its own feature drift but never on a context whose token differs from the training sequence. Prediction-shift supervision runs at exactly the training-time-test state of the second rollout step: the same drafter feature (its own output from the first step), the same cache over the prefix (built from target features), and only the input token replaced by $\hat{x}_t$. It thus supervises the token pathway that training-time test leaves unsupervised. HASS (Zhang et al., 2025) likewise aligns training with the drafter's own features, and DistillSpec (Zhou et al., 2024) distills the target distribution on drafter-generated contexts, which corresponds to our direct-distillation control.

Why $\hat{x}_t \neq$ target top-1. Responses are generated by the target, so x_t is usually the target's top-1 token, and substituting it would leave the context unchanged.

Target forward. The offline cache holds the target's hidden states and logits on the original sequence, which give z_T^{orig}. The substituted sequence $\bar{x}$ is scored by a live target forward during training, one sample at a time over its unpadded length, which gives z_T^{alt}. No hidden state from this forward reaches the drafter.

Algorithm 2 One training step with prediction-shift supervision

Read $(x,\ \mathbf{H},\ z_T^{\text{orig}})$ from the offline cache
Rollout step 1: drafter logits $z_D^{(1)}$; $\hat{x}_t \leftarrow \arg\max z_{D,t}^{(1)}$
$\mathcal{S} \leftarrow \{t : \hat{x}_t \neq \arg\max z_{T,t}^{\text{orig}}\}$
$\bar{x} \leftarrow x$ with $\bar{x}_t = \hat{x}_t$ for $t \in \mathcal{S}$; $z_T^{\text{alt}} \leftarrow \text{Target}(\bar{x})$ (no gradient)
Rollout step 2 (normal): input tokens $x \to z_D^{\text{orig}}$
Rollout step 2 (branch): same features and cache, tokens $\hat{x}$ at $\mathcal{S} \to z_D^{\text{alt}}$
$\mathcal{L}_{\text{shift}} \leftarrow \text{mean}_{t\in\mathcal{S}} \frac{1}{|V_D|}\|\Delta_{D,t} - \Delta_{T,t}\|_2^2$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{std}} + \lambda_{\text{shift}}\mathcal{L}_{\text{shift}}$

H Benchmark Details

Table 8 lists the source and split of each benchmark. For benchmarks marked "shuffle", the 80 prompts are the first 80 after shuffling with seed 42; TextVQA and VQAv2 take the first 80 records in file order.

Table 8: Benchmark sources. 80 prompts each, at most 500 generated tokens.

Benchmark	Source	Split	Selection
MMMU	`MMMU/MMMU` (Accounting, Art, Biology, Math)	test	shuffle; ≤ 3 images
SEED-Bench	LLaVA SEED-Bench question file	–	shuffle
MME	LLaVA MME question file	–	shuffle
VQAv2	`lmms-lab-encoder/VQAv2`	test	first 80
MM-Vet	`mm-vet.json`	–	shuffle
MathVista	`AI4Math/MathVista`	testmini	shuffle; image, English only
ScienceQA	`derek-thomas/ScienceQA`	validation	shuffle; image only
GQA	LLaVA GQA question file	testdev-balanced	shuffle
TextVQA	TextVQA 0.5.1	test	first 80
DocVQA	`lmms-lab-encoder/DocVQA`	test	shuffle

I Acceptance by Draft Step

Table 9 and Figure 3 break acceptance down by draft step, computed from the evaluation logs behind Tables 1 and 3. For every verification round, step k counts as accepted when at least k draft tokens are accepted; values are pooled over rounds within a benchmark and averaged over the ten benchmarks.

Table 9: $P(\text{accepted length} \geq k)$ by draft step k, greedy decoding, ten benchmarks. 20-epoch rows use prompts whose output matches autoregressive decoding; 4-epoch rows use all prompts.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
EAGLE-3 (20 ep)	0.789	0.603	0.363	0.248	0.161
HAWK (20 ep)	0.840	0.702	0.525	0.365	0.233
No auxiliary loss (4 ep)	0.767	0.585	0.344	0.214	0.136
Direct distillation (4 ep)	0.770	0.595	0.354	0.235	0.148
Prediction shift (4 ep)	0.789	0.612	0.373	0.248	0.168