To Copy or Not to Copy: Controlling Speculative Decoding via Intrinsic Model Signals

Roy Eisenstadt[1,2], **Ido Cohen**[2], **Edo Cohen-Karlik**[2]
Lior Wolf[1], **Itamar Zimerman**[1,2]

Abstract

Speculative Decoding (SD) has significantly accelerated Large Language Model (LLM) inference, yet existing approaches face a fundamental tradeoff between two drafting strategies: neural drafting and context-based copying. Neural drafts (e.g., EAGLE3) provide robust performance across diverse text settings, while copy-based methods achieve higher speedups in copy-intensive regimes by generating candidates faster and exploiting long repetition spans for near-perfect speculation. We analyze existing copy-based methods and find that they are prone to *accidental repetitions* where surface-level n-gram overlap does not reflect a structural intent to copy, leading to false-positive triggers that ultimately degrade throughput. We introduce *SwitchSD*, an adaptive framework that treats copying as a *latent control signal* of the LLM. By training lightweight probes on the target model's internal representations, SwitchSD identifies genuine *copy-intent* with high precision (AUC > 0.99). This allows the system to dynamically switch between neural drafting (e.g., EAGLE) and context-based copying. Our results across Llama and Qwen families demonstrate consistent throughput gains of up to 8.1% over state-of-the-art baselines like EAGLE3, effectively turning "copying" from a noisy heuristic into a principled, model-aware decoding regime.

1 Introduction

Large Language Models (LLMs) have redefined state-of-the-art performance across nearly all NLP domains, often exhibiting emergent reasoning abilities [Kojima et al., 2022]. However, these gains are increasingly gated by the computational cost of autoregressive decoding. This bottleneck is most acute in the *test-time scaling regime* [Snell et al., 2024, Welleck et al., 2024], where generating exhaustive chain-of-thought (CoT) sequences significantly inflates latency. As LLMs move toward deeper reasoning, the sequential nature of decoding remains the primary hurdle to real-world deployment.

Speculative decoding (SD) has emerged as the standard remedy for this bottleneck [Leviathan et al., 2023, Chen et al., 2023], utilizing a lightweight draft model to propose token sequences that the target model verifies in parallel. Beyond standard neural drafts, non-neural approaches such as *Prompt Lookup Decoding (PLD)* [Saxena, 2023] and retrieval-based speculators [Quan et al., 2025] have attempted to bypass neural overhead by copying directly from the context via n-gram search. Building on these ideas, recent innovations like *CopySpec* [Dumitru et al., 2025] introduced a hybrid heuristic: if a prefix has appeared previously in the context, the system speculates the subsequent continuation, falling back to a neural draft strategy if no previous occurrence is found. While intuitive, this approach relies on a brittle assumption that surface-level n-gram overlap always implies a structural intent to reproduce context.

Our analysis reveals a critical flaw in these surface-level heuristics: they cannot distinguish between *true copy-intent* and *accidental repetitions*. As shown in Figure 2, repeated prefixes often arise

[1]Tel Aviv University [2]Stealth AI Lab, Tel Aviv

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

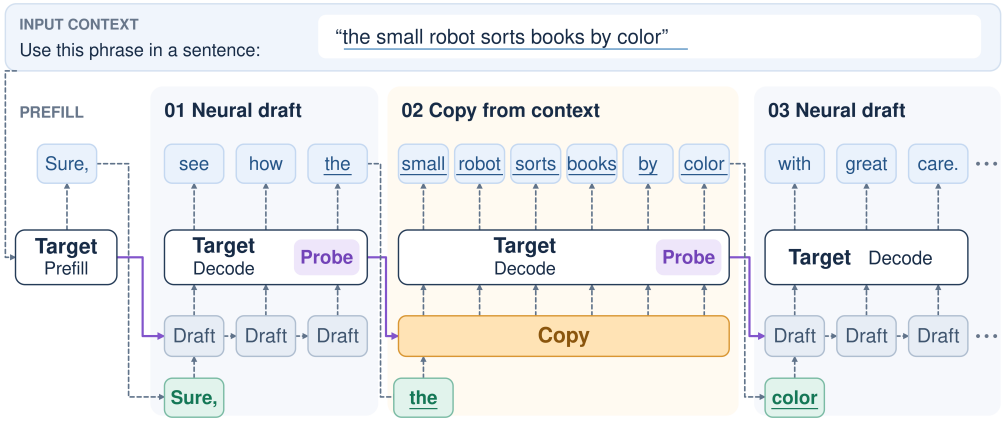


Figure 1: **Token generation process under our method.** After every verification step by the target model, given the final accepted token (shown in green), there are three possible cases: (i) No repetition contains the currently generated token. In this case, a standard neural draft model is used. Otherwise, a repetition containing the current token exists. In this case: (ii) if the probe (shown in purple) identifies copy-intent, speculative tokens are drafted by copying from the context, and (iii) if no copy-intent is identified, the neural draft model is used. The neural draft model is shown in gray, the copy draft in orange, generated tokens in blue, and repeated input tokens with underline. The example in the figure contains the three cases in their numbered order.

from coincidental associative patterns (e.g., in math reasoning) rather than a deliberate reproduction of structured text. Triggering a copy-based draft in these accidental regimes results in immediate rejection, incurring a "speculation tax" that degrades throughput compared to standard neural drafting. We argue that copying is not a surface phenomenon but is driven by a *latent control signal* encoded within the model's internal representations.

Motivated by this, we propose ***SwitchSD***, the first model-aware SD framework that replaces noisy heuristics with a principled, representation-driven decision mechanism. By extracting latent copy-signals directly from the target model's *intrinsic representations*, *SwitchSD* identifies the intent to copy which allows for dynamic orchestration: speculating via context-copying only when the model is in a "copy-regime," and falling back to state-of-the-art neural speculators (e.g., EAGLE3 [Li et al., 2025]) otherwise. This selective mechanism eliminates false-positive copy attempts, transforming context-copying from a brittle heuristic into a robust, high-yield acceleration strategy.

Our main contributions are as follows: (i) We introduce *SwitchSD*, an adaptive SD framework that dynamically switches between neural drafting and context copying using lightweight probes over the target model's hidden representations. Our approach is orthogonal to existing SD methods and consistently improves strong baselines, including EAGLE3, by up to 8.1% throughput across multiple benchmarks and model families. (ii) We show that SD contains heterogeneous decoding regimes with substantially different acceptance-length characteristics. In particular, copy-favorable regions produce significantly longer accepted continuations than standard generative regions. Exploiting this structure enables more effective draft-policy selection and improved tuning of speculative lookahead. (iii) Through extensive ablations and analysis, we demonstrate that internal model representations provide reliable control signals for adaptive SD and systematically analyze the impact of probe design, training data, threshold selection, and drafting strategies on overall decoding efficiency.

2 Background and Related Work

This section establishes the formal framework for SD and context-copying mechanisms.

Auto-Regressive Transformers. Modern LLMs [Vaswani et al., 2017] consist of stacked Transformer blocks where hidden representations H^ℓ are transformed through self-attention and MLP

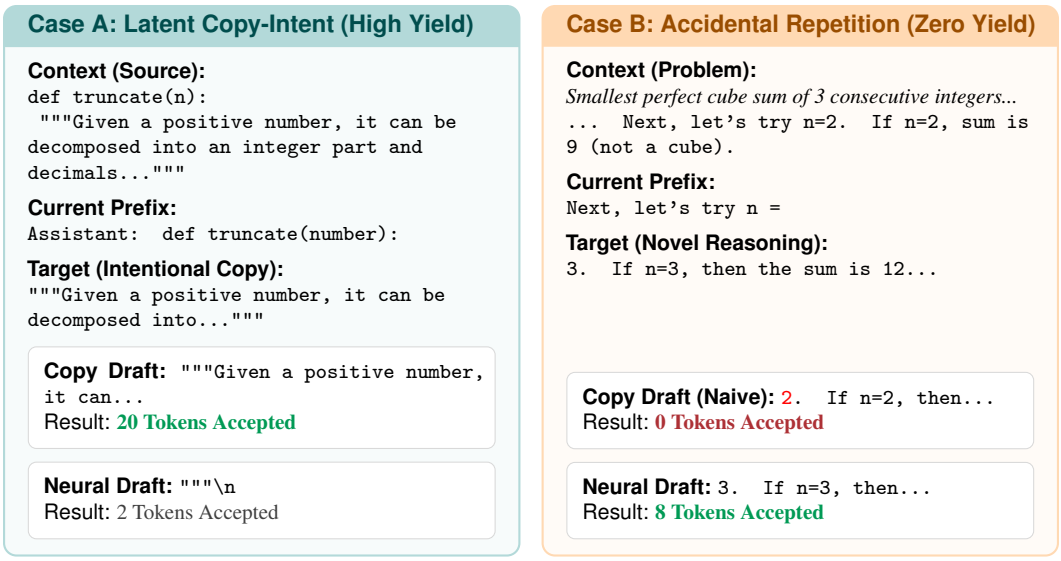


Figure 2: **Latent Copy-Intent vs. Accidental Repetition. (Left)** *SwitchSD* identifies copy-intent in docstrings. **(Right)** In reasoning tasks, n-gram overlaps are incidental; *SwitchSD* correctly defaults to neural drafting.

sublayers. Under the pre-normalization formulation, a block is defined as:

$$\begin{aligned} \tilde{H}^{\ell} &= H^{\ell} + \mathrm{Attn}\Big(\mathrm{LN}(H^{\ell})\Big), \\ H^{\ell+1} &= \tilde{H}^{\ell} + \mathrm{FFN}\Big(\mathrm{LN}(\tilde{H}^{\ell})\Big). \end{aligned} \tag{1}$$

In autoregressive LLMs, tokens x_t are generated sequentially, conditioned on the full preceding context. Our work specifically utilizes the intermediate representations H^{ℓ} at layer ℓ to detect latent decoding regimes.

Speculative Decoding. SD [Leviathan et al., 2023, Chen et al., 2023] accelerates inference by using a lightweight draft model to propose k candidate tokens, which are verified in parallel by the target model. Efficiency is governed by the acceptance rate $\alpha = 1 - KL(p, q)$, where p and q represent the target and draft distributions, and the cost ratio $c = T_p/T_q$ between the two models. While neural drafts like EAGLE [Li et al., 2024] or distillation methods [Zhou et al., 2024] improve α through alignment, they typically apply a uniform drafting policy regardless of the underlying text structure.

Context-Based and Adaptive Speculation. Beyond neural drafting, context-based methods like *Prompt Lookup Decoding (PLD)* [Saxena, 2023] use n-gram matching to propose candidates directly from the context. *CopySpec* [Dumitru et al., 2025] introduced a heuristic switch between n-gram matching and neural drafting. However, these methods remain model-agnostic and fail at "accidental repetitions." Recent work has explored statistical orchestration: *BanditSpec* [Hou et al., 2025] treats strategy selection as a multi-armed bandit problem, using UCB-based feedback on throughput to optimize drafting. Unlike these statistical "black-box" approaches that require noisy exploration phases, *SwitchSD* uses a "white-box" signal-probing internal states to identify copy-intent with high precision before speculation begins.

The Copy Mechanism in LLMs. We distinguish between memorization (training-data copying) and associative copying (contextual reproduction). The latter is driven by specific circuits known as *induction heads* [Olsson et al., 2022, Elhage et al., 2021]. Conceptually, *SwitchSD* echoes the gating logic of *Pointer-Generator Networks* [See et al., 2017], which used a learned scalar to switch between generation and pointing. However, we pivot this concept from generation-time vocabulary gating to inference-time speculation orchestration. Following evidence that task-relevant information is linearly encoded in H^{ℓ} [Park et al., 2024, Hendel et al., 2023, Eisenstadt et al., 2025], we hypothesize that copy-intent manifests as a linearly separable signal, allowing us to train lightweight probes as high-precision control signals for SD.

3 Method

3.1 Motivation: The Efficiency Tax of Accidental Repetitions

We define *accidental repetitions* as scenarios where the current context contains an n-gram overlap with the past, yet the model's target distribution p does not align with the historical continuation. As illustrated in Figure 2, these cases frequently arise from coincidental associative patterns (e.g., repeating the variable name "n" in a mathematical derivation) rather than structured reproduction.

In such instances, indiscriminate copying incurs a significant *speculation tax*: the system spends computational budget on draft proposals and parallel verification only to suffer a 0-token acceptance rate. Our analysis shows that in reasoning-heavy tasks, these false-positives trigger over *50%* of all copy attempts. This motivates the need for a *latent-aware trigger* that bypasses surface-level coincidences by directly querying the model's internal representations for genuine copy-intent.

3.2 Learning a Copy-Intent Probe

Drawing on the insight that copying constitutes a distinct decoding regime, we develop a task-agnostic procedure to probe for copy-intent directly within the model's hidden states. Unlike heuristic triggers that monitor surface-level text, our probe identifies the underlying *latent mode* of the transformer.

Problem Formulation. We formulate copy-intent detection as a binary classification task. Since surface-level repetitions are inherently noisy, we adopt a conservative proxy for our ground-truth training labels. We define a token position as a *copy-mode* instance ($Y = 1$) if it belongs to a verbatim sequence of at least $n_{\text{train}} = 5$ tokens that appears earlier in the context. This threshold ensures that our probe ignores incidental n-gram overlaps and focuses on sustained, deliberate copying behavior. All other positions are labeled as non-copy ($Y = 0$). Our objective is not to claim that *copy intent* is fully characterized by exact n-gram overlap, but rather to use longer verbatim spans as a practical indicator of an underlying latent *copy mode*. Further discussion of this matter is in Appendix E.

Controlled Probing Dataset. Because long repeated n-grams are sparse in standard pretraining corpora, we construct a *calibrated stimulus set* to isolate the copy-circuit. We utilize Claude Sonnet [Anthropic, 2025] to generate 1,000 prompts that exhibit diverse repetition structures, ranging from structured code boilerplate to repetitive linguistic templates. By having the target model generate completions for these prompts, we extract a dataset $\mathcal{D}_\ell$ of hidden representations $H_j^{(\ell)}$ paired with their corresponding copy-labels Y_j:

$$\mathcal{D}_\ell := \{(H_{i,j}^{(\ell)}, Y_{i,j}) \mid i \in [1000], j \in \{M_i, \ldots, N_i\}\}\,, \tag{2}$$

where M_i denotes the start of the decoded response. This procedure provides hundreds of thousands of labeled token-level instances, ensuring a high signal-to-noise ratio for probe training. Appendix C provides representative prompts from the dataset, illustrating their varying degrees of copy-inducing behavior.

Probe Architecture and Training. Consistent with the hypothesis that task-relevant features are linearly encoded in the residual stream [Park et al., 2024], we implement the probe as a *strictly linear projection*. For each layer ℓ, we train an independent probe $\mathcal{PR}_\ell$ that operates directly on the hidden representation $H_j^{(\ell)} \in \mathbb{R}^d$ without a bias term:

$$p_\ell(j) = \sigma(w_\ell^\top H_j^{(\ell)})\,, \tag{3}$$

where $p_\ell(j)$ represents the probability that the model is in copy-mode. Probes are trained using binary cross-entropy loss with ℓ_2 regularization. This minimal architecture ensures that probe predictions are purely a function of the vector direction in the latent space, adding negligible computational overhead. Empirically, we find that probes trained on representations extracted after the *attention sublayer* consistently outperform those from the MLP sublayers, reinforcing the connection to attention-driven induction circuits.

Layer and Threshold Selection. After training probes across all L layers, we select a single optimal layer using a greedy F1-score maximization on a held-out validation set. This identifies the specific depth at which copy-intent is most linearly separable, typically in the intermediate layers (e.g., layer 14 for Llama-3.1-8B). We then fix a decision threshold τ by maximizing the F1-score on

the same validation split. At inference time, *SwitchSD* enables copy-based drafting only when the probe score $\mathcal{PR}(H_j^{(\ell)}) \geq \tau$, providing a high-precision gate that filters out accidental repetitions.

3.3 *SwitchSD*: Copy-Intent-Aware SD

Building upon the selected probe $\mathcal{PR}$, *SwitchSD* employs a dynamic speculative framework that switches between neural drafting and context-copying strategies. At each decoding step, the system extracts the hidden representation $H_i^{(\ell)}$ and queries the probe to determine the optimal drafting regime (see Figure 1).

Copy-Intent Decision Rule. Let i denote the current token position. Copy-based speculation is enabled if and only if the latent probe score exceeds the threshold τ, *and* the current context provides structural evidence for a repetition:

$$\mathcal{PR}(H_i^{(\ell)}) \geq \tau \quad \text{where} \quad \mathcal{PR}(H_i^{(\ell)}) = \sigma(w^\top H_i^{(\ell)})\,. \tag{4}$$

To suppress accidental short-prefix matches, we further require that the κ-gram ending at the current token $(x_{i-\kappa+1}, \ldots, x_i)$ matches at least one prior occurrence in the context (we set $\kappa = 5$). If Eq. 4 holds but no κ-gram match is found, *SwitchSD* falls back to neural drafting, ensuring that speculation is only triggered when both internal intent and external structure align.

Copy-Based Speculative Generation. When the copy-regime is triggered, *SwitchSD* retrieves up to W candidate continuations, each of length D, from the context. These candidates are ranked by recency, prioritizing the most recent linguistic patterns. This yields a candidate set of plausible verbatim continuations without requiring any additional draft model parameters.

Parallel Verification with Block-Triangular Attention. To verify these W candidates efficiently, we adopt a *block-triangular attention* pattern, enabling parallel validation in a single target-model forward pass. We concatenate the W candidates sequentially after the anchor position. The attention mask is structured such that tokens within a specific candidate $w \in [W]$ can attend to: (i) the full original prefix, and (ii) preceding tokens within candidate w itself, while remaining isolated from other candidates. This allows the target model to verify all hypotheses simultaneously. *SwitchSD* then selects the longest validated continuation for the next decoding step. A controlled component ablation in App. F shows that parallel multi-candidate verification is a major contributor to the resulting throughput gains.

Non-Copy Speculative Tokens. In generative regimes where copy-intent is not detected, *SwitchSD* delegates drafting to traditional SD mechanisms. To demonstrate the framework's flexibility, we evaluate *SwitchSD* using both vanilla *Speculative Sampling (SPS)* and the state-of-the-art *EAGLE3* [Li et al., 2024]. This "orchestrator-first" design proves that signal-aware switching provides benefits that are orthogonal to the specific choice of a neural draft model.

KV Cache Synchronization. Orchestrating between copy-based and neural-based paths requires careful coordination of Key-Value (KV) caches. When a copy-based sequence is accepted, the draft model's KV cache may lag behind. *SwitchSD* resolves this via *lazy propagation*: accepted tokens are passed through the draft model during the subsequent forward pass. By merging this synchronization with the next drafting step, the synchronization cost is effectively *amortized*, ensuring that both the target and draft caches remain coherent with minimal latency overhead.

Overall Framework. *SwitchSD* acts as an orchestration layer over SD: at each step, it dynamically selects between copy-based and draft-based speculation, while delegating final verification to the target model. This design preserves correctness guarantees and is fully compatible with both classical speculative sampling and state-of-the-art methods such as *EAGLE3*.

4 Experiments

This section presents a systematic evaluation of *SwitchSD*. We begin by reporting main results across multiple benchmarks, comparing *SwitchSD* against heuristic, statistical, and neural baselines (§4.1). We then provide a detailed model analysis to characterize how latent signals improve acceptance lengths (§4.2). Finally, we justify our design choices through dedicated ablation studies (§4.3).

Table 1: Throughput and speedup relative to vanilla decoding across datasets and models on H200 GPUs. Results report mean ± sample standard deviation over three runs. For CopySpec and *SwitchSD*, Copy and Neural statistics report the percentage of speculation iterations routed to each path and its mean accepted length.

Dataset	Method	LLaMA-3.1-8B-Instruct						LLaMA-3.3-70B-Instruct						Qwen3-8B					
		Tok/s	Speedup	Copy		Neural		Tok/s	Speedup	Copy		Neural		Tok/s	Speedup	Copy		Neural	
				%	mean	%	mean			%	mean	%	mean			%	mean	%	mean
CNN/DM	Vanilla	57.89±0.54	1.00×	–	–	–	–	14.1±0.3	1.00×	–	–	–	–	41.71±0.61	1.00×	–	–	–	–
	PLD	78.76±0.44	1.36×	–	–	–	–	20.2±0.5	1.43×	–	–	–	–	49.39±0.21	1.18×	–	–	–	–
	BanditSpec	73.19±0.05	1.26×	–	–	–	–	13.8±0.2	0.98×	–	–	–	–	61.20±0.07	1.46×	–	–	–	–
	CopySpec	82.43±0.79	1.42×	11.0%	0.64	89.0%	2.54	30.4±1.0	2.16×	8.6%	0.68	91.4%	2.59	61.73±7.31	1.48×	8.7%	1.75	91.3%	1.85
	EAGLE3	88.0±0.6	1.52×	–	–	100%	2.57	31.8±0.3	2.25×	–	–	100%	2.61	67.67±0.94	1.62×	–	–	100%	1.97
	SwitchSD	**89.2±0.5**	**1.54**×	7.1%	1.44	92.9%	2.53	**32.1±0.4**	**2.28**×	5.3%	2.87	94.8%	2.58	**71.24±0.49**	**1.71**×	2.4%	2.38	97.6%	1.90
Math500	Vanilla	57.61±0.46	1.00×	–	–	–	–	14.6±0.2	1.00×	–	–	–	–	41.84±0.87	1.00×	–	–	–	–
	PLD	81.94±0.69	1.42×	–	–	–	–	20.3±0.9	1.39×	–	–	–	–	56.46±0.52	1.35×	–	–	–	–
	BanditSpec	84.07±0.82	1.46×	–	–	–	–	14.3±1.4	0.98×	–	–	–	–	68.51±1.14	1.63×	–	–	–	–
	CopySpec	86.03±10.28	1.49×	27.1%	2.18	72.9%	2.58	34.0±0.7	2.33×	22.2%	1.57	77.8%	2.83	66.61±1.82	1.59×	14.3%	1.40	85.7%	2.23
	EAGLE3	97.17±1.04	1.69×	–	–	100%	2.81	37.4±1.0	2.56×	–	–	100%	2.96	71.51±0.14	1.71×	–	–	100%	2.24
	SwitchSD	**101.8±2.6**	**1.77**×	16.9%	4.83	83.1%	2.57	**37.5±0.4**	**2.57**×	6.5%	4.42	93.5%	2.88	**77.32±0.82**	**1.85**×	8.9%	2.71	91.1%	2.11
HumanEval	Vanilla	58.22±0.49	1.00×	–	–	–	–	14.6±0.1	1.00×	–	–	–	–	42.24±0.94	1.00×	–	–	–	–
	PLD	82.57±2.99	1.42×	–	–	–	–	21.0±1.2	1.44×	–	–	–	–	55.57±0.30	1.32×	–	–	–	–
	BanditSpec	78.52±0.81	1.35×	–	–	–	–	15.3±0.8	1.05×	–	–	–	–	80.99±1.36	1.91×	–	–	–	–
	CopySpec	117.87±0.49	2.02×	28.1%	2.32	71.9%	3.78	44.7±0.6	3.06×	20.4%	1.96	79.6%	4.05	76.03±2.13	1.80×	17.4%	1.93	82.6%	2.50
	EAGLE3	122.86±0.80	2.11×	–	–	100%	4.09	48.7±0.8	3.34×	–	–	100%	4.21	81.09±1.05	1.92×	–	–	100%	2.52
	SwitchSD	**132.69±0.55**	**2.28**×	16.8%	5.48	83.2%	3.77	**49.6±1.1**	**3.40**×	12.4%	4.78	87.6%	4.10	**83.46±0.11**	**1.98**×	6.9%	3.16	93.1%	2.44

Additional experiments analyzing the impact of temperature sampling are provided in App. A, showing that *SwitchSD* maintains SoTA performance even at high temperatures, and additional comparisons against context-exploitation baselines are presented in App. B.

4.1 Main Result

We evaluate *SwitchSD* across three primary domains: coding (HumanEval [Chen et al., 2021]), mathematical reasoning (Math500 [Hendrycks et al., 2021]), and summarization (CNN/DailyMail [Hermann et al., 2015]). Performance is measured by decoding throughput (tokens per second) and the resulting speedup relative to vanilla autoregressive decoding.

Baselines & Models. We compare *SwitchSD* against three classes of speculative decoding: (i) *Heuristic-based: PLD* [Saxena, 2023] (naive n-gram search) and *CopySpec* [Dumitru et al., 2025] (prefix-based switching). (ii) *Neural-only: EAGLE3* [Li et al., 2025], a strong neural speculative drafting method. (iii) *Statistical-Adaptive: BanditSpec* [Hou et al., 2025], which uses a UCB-based bandit to switch between strategies based on throughput rewards. For model selection, we conduct experiments using models that are already supported by EAGLE3. In particular, we evaluate models from the LLaMA-3 [Grattafiori et al., 2024] and Qwen-3 [Yang et al., 2025] families. App. D contains full implementation and reproducibility details.

Throughput and Speedup. Across all 9 configurations in Table 1, *SwitchSD* consistently achieves the highest mean throughput, matching or outperforming all baselines. The gains are most pronounced on the 8B models: *SwitchSD* improves over EAGLE3 by 8.0%, 4.7%, and 1.4% on HumanEval, Math500, and CNN/DailyMail with LLaMA-3.1-8B, and by 2.9%, 8.1%, and 5.3% with Qwen3-8B, respectively. Across these six configurations, the mean ± standard-deviation intervals of *SwitchSD* and EAGLE3 do not overlap. These trends hold even when using a weaker draft model (SPS) on Qwen3-8B (Table 2), where *SwitchSD* continues to dominate.

Table 2: Decoding throughput and speedup for Qwen3-8B (8-bit) using SPS with a Qwen3-0.6B draft model. Copy and SPS statistics report % iterations and mean acceptance length.

Dataset	Method	Tok/s	Speedup	Copy		SPS	
				%	mean	%	mean
CNN/DM	Vanilla	7.53	1.00×	–	–	–	–
	CopySpec	10.30	1.37×	5.8	1.77	94.2	2.46
	SPS	9.74	1.29×	–	–	100	2.72
	SwitchSD	**10.43**	**1.38**×	3.2	2.79	96.8	2.47
Math500	Vanilla	7.84	1.00×	–	–	–	–
	CopySpec	13.18	1.68×	17.7	2.05	82.3	3.09
	SPS	13.47	1.72×	–	–	100	3.77
	SwitchSD	**13.85**	**1.77**×	8.6	4.08	91.4	3.10
HumanEval	Vanilla	7.87	1.00×	–	–	–	–
	CopySpec	12.31	1.56×	14.3	2.08	85.7	2.88
	SPS	12.35	1.57×	–	–	100	3.39
	SwitchSD	**13.09**	**1.66**×	7.2	4.37	92.8	2.89

Internal Signals vs. Statistical Exploration. A key finding is the contrast between *SwitchSD* and *BanditSpec*. While BanditSpec adaptively selects between drafting strategies based on observed through-

put rewards, this requires exploration of potentially inefficient copy actions. The effect is particularly large on LLaMA-3.3-70B: on Math500, BanditSpec achieves only $0.98\times$ speedup over vanilla decoding, whereas *SwitchSD* reaches $2.57\times$. Similar behavior appears on CNN/DailyMail, where BanditSpec remains below vanilla throughput ($0.98\times$) while *SwitchSD* reaches $2.28\times$. By querying the target model's hidden state before selecting a drafting strategy, *SwitchSD* avoids this exploration cost and uses copying only when the latent signal indicates that it is likely to yield a useful continuation. This highlights a central advantage of intrinsic model signals over reward-based online strategy selection: routing decisions can be made directly from the model state rather than learned through costly trial and error during decoding.

High-Yield Speculation. The path-level statistics in Table 1 show that *SwitchSD* gains efficiency by making copying more selective and higher-yield. For example, on LLaMA-3.1-8B HumanEval, CopySpec routes 28.1% of speculation iterations through the copy path with a mean accepted length of 2.32 tokens, whereas *SwitchSD* copies on only 16.8% of iterations while accepting 5.48 tokens on average. The same pattern persists at larger scale: on LLaMA-3.3-70B HumanEval, *SwitchSD* copies on 12.4% of iterations with a mean accepted length of 4.78, compared with 20.4% at 1.96 for CopySpec. Thus, the intrinsic signal does not improve throughput by invoking copying more aggressively; instead, it suppresses low-yield accidental repetitions and concentrates copy-based speculation on substantially more productive decoding steps.

Robustness Across SD Architectures. To evaluate whether the benefits of *SwitchSD* are tied to specific neural SD architectures, we report results for Qwen3-8B using standard *Speculative Sampling (SPS)* with a 0.6B draft model in Table 2. Unlike EAGLE3, which predicts future feature states, SPS relies on token-level sampling from a smaller model.

Despite this shift in the underlying neural speculator, *SwitchSD* consistently achieves the highest throughput across all benchmarks. For example, on HumanEval, it increases throughput to 13.09 Tok/s, outperforming both the base SPS (12.35) and the heuristic-based CopySpec (12.31). These results confirm that signal-aware orchestration is *draft-agnostic*; the ability to identify a "copy-regime" via internal representations provides a significant efficiency layer that is orthogonal to the strength or type of the neural draft model.

Precision Advantage Carries Across SD Architectures. A comparative analysis of the iteration statistics in Table 2 reveals the fundamental mechanism behind our gains. Across all datasets, *SwitchSD* invokes copying significantly more *selectively* than the heuristic CopySpec, yet achieves *longer accepted spans*. This "less is more" pattern is the mathematical signature of our latent copy-intent probe: (i)*HumanEval: SwitchSD* triggers context-copying in only 7.2% of steps (vs. 14.3% for CopySpec), yet yields a mean acceptance length of 4.37 (vs. 2.08). (ii) *Math500:* We trigger copying in only 8.6% of steps (vs. 17.7%), but accept 4.08 tokens on average (vs. 2.05).

By filtering out the "accidental repetitions" where n-gram overlaps exist but the model's actual intent remains generative, *SwitchSD* ensures that context-copying is only utilized when it can provide high-yield leaps. This precision allows the neural draft model to focus on the generative regions where it excels, leading to a more efficient division of labor in the decoding pipeline.

Task-Specific Performance. The performance gains are larger in structured domains like coding and mathematics. In these tasks, the model transitions between rigid, repetitive logic (copy-heavy) and novel reasoning (generative). *SwitchSD* successfully navigates these transitions by "listening" to the model's internal state. In contrast, on CNN/DailyMail, where the repetition is naturally weaker, the gains are more modest but still favor the signal-aware approach. Together, these results demonstrate that *SwitchSD* acts as a robust, universal orchestrator that optimizes the interplay between contextual and neural speculation.

4.2 Model Analysis

4.2.1 Regime Separation and Acceptance Yield

In this section, we provide a fine-grained analysis of *SwitchSD*'s internal mechanics, demonstrating how latent-signal probing successfully partitions decoding into two distinct mathematical regimes.

To quantify the probe's ability to identify high-yield sequences, we perform a post-hoc analysis on HUMANEVAL using Llama-3.1-8B-Instruct. We execute both the context-copying and EAGLE3 paths at every step, partitioning the results based on the probe's prediction. Figure 3 illustrates the resulting probability mass functions for speculative acceptance lengths. The distributions reveal a near-total separation of decoding modes. When the probe is *inactive* (blue), the copy-based draft suffers from a heavy concentration of short acceptances (0–2 tokens), representing the "accidental repetition" regime where copying is inefficient. Conversely, when the probe is *active* (red), the distribution shifts dramatically toward high-yield runs. Most notably, *30.0%* of trajectories reach the maximum speculative length of 20 tokens when the probe is active, compared to a mere *4.7%* when it is inactive. This *6.3*$\times$ increase in maximum-length yield confirms that our probe identifies a latent state of high predictability that surface-level heuristics cannot detect. These results suggest that the probe captures an internal copy-intent signal rather than simply detecting surface-level repetition patterns.

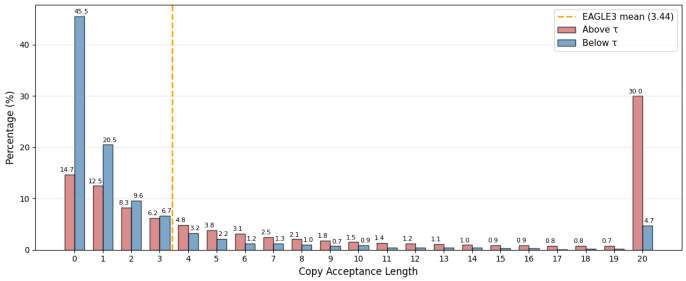


Figure 3: **Copy acceptance by probe prediction.** Accepted copy-draft tokens with the probe active (red) or inactive (blue).

4.2.2 Impact on Optimal Lookahead (γ)

The partitioning of decoding into "copy" and "generative" regimes has significant implications for the neural draft's optimal lookahead, γ. Typically, γ is tuned based on the average acceptance rate across all tokens. However, context-copying tokens are inherently easier to speculate, which artificially inflates the global acceptance rate. By routing these high-probability tokens through the copy path, *SwitchSD* removes the "easy" examples from the neural draft's workload. As a result, the remaining tokens assigned to the neural speculator are harder on average, leading to the slightly lower optimal γ values observed in Table 3 (e.g., a shift from 7 to 5 for Llama-3.3-70B).

Table 3: Best draft lookahead γ selected on the evaluation set for EAGLE3 and *SwitchSD*.

Approach	LLaMA-3.1-8B	Qwen3-8B	LLaMA-3.3-70B
EAGLE3	7	5	7
SwitchSD	6	5	5

This separation enables a critical *system-level optimization*: instead of over-speculating on generative tokens with a high global γ, *SwitchSD* allows for targeted tuning. By using a shorter, more appropriate lookahead for the neural draft and a high-yield span for the copy draft, we eliminate computational waste and maximize the throughput of both paths. This refined resource allocation is a primary driver of the speedups reported in §4.1.

4.3 Ablation Studies

We conduct a series of ablations to isolate the impact of our training data, representational placement, and decision thresholds on *SwitchSD*'s performance. We additionally disentangle the system-level contributions of probe-based routing and parallel multi-candidate verification through a controlled six-arm ablation in App. F.

4.3.1 Importance of Intent-Driven Training Data

To evaluate our synthetic data construction, we compare the efficacy of probes trained on three distinct data distributions: (i) *ConstructedCopy*, a non-semantic dataset of repeated random token spans; (ii) *WikiText-103*, representing natural linguistic repetition without explicit prompt-completion structure; and (iii) *CopyDiversity* (our method), which utilizes model-generated completions across diverse repetition templates.

As shown in Figure 4, the non-semantic *ConstructedCopy* yields the weakest performance ($F1 = 0.65$), confirming that copy-intent is not merely a structural phenomenon but is tied to the model's semantic processing. While *WikiText-103* improves performance to 0.81, it remains significantly below our full method (0.87). The confusion matrices illustrate this gap: probes trained on natural text or random spans suffer from higher false-positive rates on MT-BENCH, likely failing to distinguish between coincidental reuse and intentional reproduction. These results underscore that high-precision probing requires *intent-driven* supervision that aligns with the target model's actual behavior.

4.3.2 Architectural Placement: Layer and Sublayer Selection

A critical design choice is the selection of internal representations. We evaluate probes trained on representations extracted from every layer at two positions: (i) after the self-attention sublayer (X') and (ii) after the MLP sublayer (Y). Figure 5 reveals two key insights. First, probes trained on *attention representations* consistently outperform those from MLP sublayers. This result aligns with our hypothesis in §2, suggesting that copy-intent is primarily carried by attention-driven induction circuits rather than feed-forward transformations. Second, the signal strength peaks in the *intermediate layers* (peaking at layer 14 for Llama-3.1-8B), while declining in the final layers as representations transform into vocabulary logits. This validates our choice of intermediate attention-based hidden states as the optimal control signal.

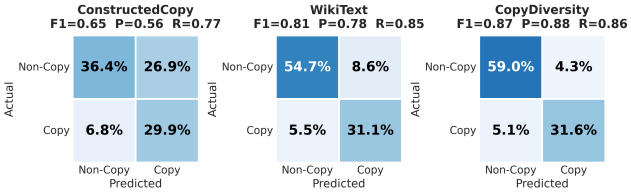


Figure 4: Confusion matrices for copy-intent probes evaluated on MT-Bench at their selected decision thresholds. Cells report the percentage of tokens from the total evaluation set.

4.3.3 Threshold Selection and Linear Separability

Finally, we analyze the impact of the decision threshold τ. Figure 6 (right) demonstrates that an F1-optimal threshold lies near 0.4 for the Llama-3.1-8B model. The corresponding ROC curve (Figure 6, left) exhibits an $AUC > 0.99$, indicating near-perfect linear separability of the copy-mode. This exceptional clarity suggests that copy-intent is a *first-order feature* in the model's residual stream. Such high separability allows *SwitchSD* to operate as a high-precision gate, achieving the significant speedups reported in §4.1 with negligible risk of false-positive copy triggers.

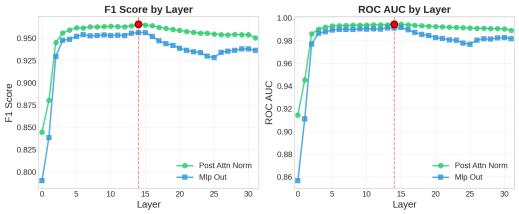


Figure 5: **Position and layer selection for probing.** F1 (left) and ROC–AUC (right) across transformer layers, using representations after attention (green) or the MLP (blue).

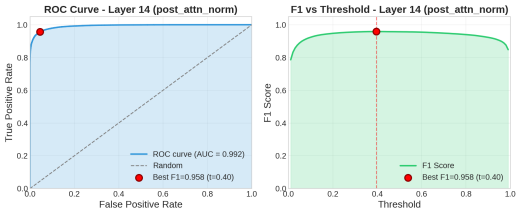


Figure 6: **Threshold selection for probing.** ROC curve (left) and F1 versus decision threshold (right) for probes on the selected representation.

5 Conclusions

We introduced *SwitchSD*, a framework that establishes a new direction for *latent-aware orchestration* in speculative decoding. By moving beyond surface-level n-gram heuristics and statistical bandits, we demonstrate that a model's internal representations provide a high-precision, "white-box" control signal for switching between decoding regimes. Across diverse model families (Llama-3, Qwen) and benchmarks (Math500, HumanEval, CNN/Dailymail), *SwitchSD* consistently delivers significant throughput gains, outperforming state-of-the-art neural speculators like EAGLE3 by up to 8.1% while preserving exact generation quality. Our findings suggest that the future of efficient LLM inference lies in *model-aware systems* that exploit the intrinsic representational states of the transformer to optimize the division of labor between neural and contextual speculation.

Furthermore, our binary copy-intent probe could be extended into a *multi-modal orchestrator* capable of predicting optimal speculation lengths (γ) or selecting between multiple specialized draft models based on latent task-type signals. Finally, we believe that deeper mechanistic interpretability of the "copy-circuits" identified in this work could lead to even more refined probes, further closing the gap between the theoretical understanding of transformer internals and the practical requirements of low-latency inference.

References

Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

Razvan-Gabriel Dumitru, Minglai Yang, Vikas Yadav, and Mihai Surdeanu. Copyspec: Accelerating llms with speculative copy-and-paste without compromising quality. *arXiv preprint arXiv:2502.08923*, 2025.

Roy Eisenstadt, Itamar Zimerman, and Lior Wolf. Overclocking llm reasoning: Monitoring and controlling thinking path lengths in llms. *arXiv preprint arXiv:2506.07240*, 2025.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *ArXiv*, abs/1506.03340, 2015. URL https://api.semanticscholar.org/CorpusID:6203757.

Yunlong Hou, Fengzhuo Zhang, Cunxiao Du, Xuan Zhang, Jiachun Pan, Tianyu Pang, Chao Du, Vincent Y. F. Tan, and Zhuoran Yang. Banditspec: Adaptive speculative decoding via bandit algorithms. *ArXiv*, abs/2505.15141, 2025. URL https://api.semanticscholar.org/CorpusID:278782841.

Yuxuan Hu, Ke Wang, Xiaokang Zhang, Fanjin Zhang, Cuiping Li, Hong Chen, and Jing Zhang. Sam decoding: Speculative decoding via suffix automaton. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12187–12204, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

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.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR, 2024.

Guofeng Quan, Wenfeng Feng, Chuzhan Hao, Guochao Jiang, Yuewei Zhang, and Hao Henry Wang. RASD: Retrieval-augmented speculative decoding. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6167–6177, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.320. URL https://aclanthology.org/2025.findings-acl.320/.

Apoorv Saxena. Prompt lookup decoding, November 2023. URL https://github.com/apoorvumang/prompt-lookup-decoding/.

A. See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Annual Meeting of the Association for Computational Linguistics*, 2017. URL https://api.semanticscholar.org/CorpusID:8314118.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

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

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yao Zhao, Zhitian Xie, Chen Liang, Chenyi Zhuang, and Jinjie Gu. Lookahead: An inference acceleration framework for large language model with lossless generation accuracy. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6344–6355, 2024.

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, 2024. URL https://arxiv.org/abs/2310.08461.

A Impact of Sampling Temperature

All main experiments in the paper are conducted with temperature $T = 0.0$, following the standard setup commonly used in SD evaluations Leviathan et al. [2023], Chen et al. [2023]. Since SD performance depends on the agreement between the draft and target model distributions, increasing the sampling temperature generally reduces acceptance rates and therefore lowers the achievable speedup.

Temperature	Base Tok/s	EAGLE3 Tok/s	SwitchSD Tok/s	Speedup (SwitchSD)
0.0	38.3	87.5	**98.6**	**2.57**×
0.2	36.3	83.1	**88.5**	**2.44**×
0.4	36.7	80.6	**83.9**	**2.29**×
0.6	36.2	79.7	**80.6**	**2.22**×
0.8	36.4	73.3	**74.9**	**2.05**×
1.0	36.8	53.9	**56.5**	**1.54**×

Table 4: Throughput across different sampling temperatures on HumanEval using LLaMA-3.1-8B.

To evaluate the robustness of *SwitchSD* under different decoding configurations, we additionally measure throughput across a range of sampling temperatures on HumanEval using LLaMA-3.1-8B-Instruct. Results are reported in Table 4.

Overall, *SwitchSD* consistently outperforms EAGLE3 across all evaluated temperatures. As expected, the performance gap gradually decreases as temperature increases. Higher temperatures induce more stochastic and diverse generations, reducing repetition frequency and lowering speculative token acceptance rates for both copy-based and neural drafting strategies. Nevertheless, even at $T = 1.0$, *SwitchSD* still achieves a substantial $1.54\times$ speedup over vanilla decoding while maintaining an advantage over EAGLE3.

B Additional Comparison to Context-Exploitation Baselines

To further evaluate the effectiveness of *SwitchSD* against non-parametric context-exploitation methods, we compare our method against several additional baselines that rely on context copying and n-gram retrieval mechanisms, including Prompt Lookup Decoding (PLD) [Saxena, 2023], Sam decoding (SAMD) [Hu et al., 2025], and Lookahead Decoding [Zhao et al., 2024]. All experiments are conducted using LLaMA-3.1-8B-Instruct.

Dataset	Method	Tok/s	Speedup	MAL (copy/EAGLE)
HumanEval	PLD	53.0	1.38×	1.59
	SAMD	48.4	1.26×	1.63
	Lookahead	77.7	2.03×	3.17
	SwitchSD	**98.63**	**2.58**×	**5.88 / 2.93**
Math500	PLD	53.1	1.37×	1.54
	SAMD	52.3	1.35×	1.60
	Lookahead	80.3	2.07×	3.10
	SwitchSD	**88.74**	**2.29**×	**5.39 / 2.41**
CNN/DailyMail	PLD	50.7	1.33×	1.51
	SAMD	44.7	1.17×	1.53
	Lookahead	63.9	1.67×	2.85
	SwitchSD	**77.00**	**2.01**×	**2.84 / 2.46**

Table 5: Comparison against additional context-exploitation SD baselines using LLaMA-3.1-8B-Instruct. MAL denotes mean accepted length.

Results are presented in Table 5. Across all datasets, *SwitchSD* consistently achieves the highest throughput and speedup. In particular, while PLD and SAMD often produce short accepted copy spans, and Lookahead improves acceptance through more aggressive retrieval strategies, *SwitchSD* obtains substantially longer accepted copy continuations while preserving the benefits of neural drafting through EAGLE3. These results further demonstrate that selectively activating copying based on intrinsic copy-intent signals is substantially more effective than indiscriminate context exploitation.

C CopyDiversity Dataset

CopyDiversity is a curated dataset designed to capture a wide spectrum of copy behavior in language model generation, ranging from strict verbatim reproduction to fully open-ended responses. The dataset contains 800 training samples spanning 17 prompt types, intentionally constructed to vary the degree to which copying from the input context is expected or beneficial. This diversity enables systematic analysis of copy intent and accidental repetition under realistic generation conditions. The dataset consists of 800 train examples and 200 validation examples.

Heavy Copy prompts (10.6% of the data) explicitly require verbatim reproduction of provided content. Representative examples include:

- *JSON Echo*: *User profile:*

  ```
  {
    "name": "Bob",
    "age": 45,
    "city": "London",
    "skills": ["SQL", "JavaScript"]
  }
  ```

  Output the same JSON with proper formatting. Copy the JSON exactly.

- *Quote Extraction*: *"Text: The Nile River is the longest river in the world. It flows through 11 countries in northeastern Africa. Gravity is the force that attracts objects toward each other. On Earth, it gives weight to physical objects. Photosynthesis converts sunlight into chemical energy. Plants use this process to produce glucose and oxygen. Identify and quote exactly 3 important sentences from the passage:"*

In these settings, faithful copying is essential for correctness.

Medium Copy prompts (11.7% of the data) require partial reuse of the input content, typically combined with light transformation or selective extraction. Example prompts include:

- *Definition Recall*:

  ```
  Key terms:
  Graph: A collection of nodes (vertices) connected by edges representing
  relationships between elements.
  Algorithm: A step-by-step procedure for solving a problem or accomplishing
  a task. Interpreter: A program that executes source code line by line without
  prior compilation.
  Linked List: A linear data structure where elements are stored in nodes
  connected by pointers.
  Server: A computer program or device that provides services and resources to
  other programs or devices.

  Define each term using the exact definitions above:
  1. What is a graph?
  2. What is an algorithm?
  3. What is an interpreter?
  4. What is a linked list?
  5. What is a server?
  ```

- *Summarize and Quote*: *"Article: The Amazon rainforest produces 20% of the world's oxygen. It is home to millions of species. The Pacific Ocean is the largest and deepest ocean on Earth. It covers more than 60 million square miles. Summarize briefly, then include 2 exact quotes from the passage."*

These tasks exhibit mixed behavior, where copying is necessary but must be integrated with generation.

Light or No Copy prompts (77.7% of the data) are open-ended and do not require verbatim reuse of the input. Examples include:

- *Creative Writing*: *Describe a world where dreams can be shared between people.*
- *Reasoning Problem*: *A lily pad doubles in size every day. If it takes 48 days to cover a lake, on which day is the lake half covered?*

In these cases, copying is generally incidental and often undesirable.

By explicitly balancing prompt types across these three regimes, CopyDiversity supports fine-grained evaluation of copy intent detection and copy-aware decoding strategies across diverse and realistic use cases.

D Reproducibility and Implementation Details

This section provides implementation and evaluation details for *SwitchSD* and the speculative decoding baselines. Unless otherwise stated, all methods within a reported comparison are executed in the same evaluation harness and under identical hardware and decoding conditions.

D.1 SwitchSD Configuration

Copy-Intent Probe. For each target model, we train linear probes on frozen hidden representations extracted after the self-attention sublayer, as described in §3.2. Each probe consists of a single bias-free linear projection followed by a sigmoid. Probe training uses binary cross-entropy with ℓ_2 regularization and operates on cached target-model representations; consequently, no target-model parameters are updated. Training the probe from cached representations takes less than two minutes on a single GPU.

We train candidate probes at multiple transformer layers and select the layer maximizing validation F1. The decision threshold τ is subsequently selected by maximizing F1 on the held-out validation split and is fixed before downstream benchmark evaluation. For example, the selected representation for LLaMA-3.1-8B-Instruct is the post-attention state at layer 14. No benchmark-specific threshold tuning is performed.

Copy Trigger. At inference time, the copy path is considered only when the selected probe satisfies

$$\mathcal{PR}(H_i^{(\ell)}) \geq \tau.$$

We additionally require a matching κ-gram in the preceding context, with $\kappa = 5$ in all main experiments. If either condition fails, decoding is delegated to the neural speculative drafter.

Candidate Retrieval and Verification. When copying is activated, *SwitchSD* retrieves up to $W = 10$ historical candidate continuations, ranked by recency. Each candidate contains at most $D = 20$ speculative tokens. Candidates are verified simultaneously by the target model using the block-triangular attention mask described in §3.3, which prevents information flow between different candidate branches while allowing each candidate to attend to the common prefix and its own preceding tokens. The longest valid continuation is committed after target-model verification.

KV-Cache Synchronization. The target and neural-drafter KV caches are synchronized lazily. When tokens from the copy path are accepted, the corresponding neural-drafter states are populated during its next invocation rather than through a separate synchronization pass. All reported throughput measurements include this synchronization cost.

D.2 EAGLE3 Configuration and Baseline Alignment

Our EAGLE3 baseline follows the official EAGLE3 implementation and uses its tree-based proposal and verification procedure. For LLaMA-3.1-8B-Instruct, we use the released `yuhuili/EAGLE3-LLaMA3.1-Instruct-8B` draft checkpoint. The proposal tree uses 10 expansion nodes and a total draft-tree budget of 60 tokens, with the speculative configuration tuned following the EAGLE3 evaluation protocol.

Importantly, EAGLE3 and *SwitchSD* are evaluated within the same implementation harness. The neural path of *SwitchSD* invokes the same EAGLE3 proposal, tree verification, and KV-cache code

path as the standalone EAGLE3 baseline. Disabling the copy path in *SwitchSD* therefore recovers the EAGLE3 baseline directly. This controls for harness-level differences when measuring the incremental contribution of copy-aware orchestration.

D.3 Evaluation Protocol

Benchmarks. The main evaluation covers HumanEval, Math500, and CNN/DailyMail, representing code generation, mathematical reasoning, and summarization, respectively. The primary EAGLE3 experiments evaluate LLaMA-3.1-8B-Instruct, LLaMA-3.3-70B-Instruct, and Qwen3-8B. Unless otherwise specified, main experiments use greedy decoding ($T = 0$). Sampling-temperature experiments are reported separately in App. A.

Repeated Runs and Hardware. The final main comparison in Table 1 is evaluated on NVIDIA H200 GPUs. Every reported configuration in this table is repeated three times under the same hardware and software environment, and throughput is reported as the mean $\pm$ sample standard deviation.

Additional experiments are performed on H100 GPUs unless otherwise stated. The controlled component ablation in App. F is evaluated on H200 GPUs to match the repeated-run main evaluation. All comparisons within an individual table or experiment use matched hardware and execution conditions.

Throughput Measurement. We report end-to-end decoding throughput in generated tokens per second. Timing includes proposal generation, probe evaluation, n-gram lookup, candidate retrieval, attention-mask construction, target-model verification, and required KV-cache synchronization. Thus, the reported speedups measure realized wall-clock performance rather than an acceptance-length proxy or an overhead-free estimate.

D.4 Switching Overhead

To quantify the additional system cost introduced by *SwitchSD*, we profile the switching path using CUDA events on LLaMA-3.1-8B-Instruct with HumanEval. The target-model forward pass in this setting takes approximately 25.7 ms.

Component	Time / step	% of target forward
Linear probe projection	$139.8\,\mu$s	0.54%
n-gram lookup	$23.6\,\mu$s	0.09%
Candidate retrieval + mask assembly	$80.6\,\mu$s	0.31%
Lazy KV-cache synchronization	$12.0\,\mu$s	0.05%
Total switching overhead	$\mathbf{273.7}\,\mu$s	$\approx \mathbf{1.06}\%$

Table 6: **Per-step switching overhead.** CUDA-event profiling on LLaMA-3.1-8B-Instruct with HumanEval. Percentages are relative to a 25.7 ms target-model forward pass. The component measurements sum to less than the reported total because the total additionally captures orchestration overhead surrounding the individually instrumented operations.

The complete switching mechanism therefore adds approximately 1.06% of the cost of a target forward pass. In particular, the linear probe itself accounts for only 0.54%. Since the throughput results throughout the paper are measured end-to-end, all reported gains over EAGLE3 are net of these costs rather than theoretical savings before orchestration overhead.

D.5 Output Preservation

SwitchSD changes only how speculative proposals are constructed; it does not bypass target-model verification. Every proposed token, whether produced by EAGLE3 or retrieved from the existing context, is verified by the target model according to the underlying speculative decoding procedure before being committed. Consequently, switching between drafting mechanisms does not alter the target model's output distribution. In particular, preferentially invoking the copy path in copy-oriented

regimes cannot itself bias the generated text toward repetition: an incorrect copied continuation is rejected by the target model in exactly the same manner as an incorrect neural draft.

E Discussion: What Does the Probe Learn?

Learning copy intent is inherently challenging, since there is no direct ground-truth annotation for this behavior. We therefore train the probe with weak supervision: automatically identified long repetitions are used as proxy labels for copy-oriented behavior. A natural concern is that the resulting probe may learn surface-level repetition rather than the copy-intent signal it is designed to capture. Our design addresses this concern in three ways:

(i) Weak supervision. Repeated spans are used only as a supervisory proxy, not as the definition of copy intent. Long verbatim continuations provide high-precision examples of situations in which copying is likely to be useful.

(ii) Representational Bottleneck. While verbatim spans provide the supervisory labels, the probe's input is strictly constrained to the hidden representation of a single token at a single timestep. This creates a significant information bottleneck: the probe does not have access to the speculative buffer or the context window as a sequence. Consequently, the probe cannot be performing a post-hoc string match. To achieve high precision, it must isolate a specialized "copy-intent" signature distilled within the latent space, a signal that indicates the model has transitioned its internal logic from novel synthesis to contextual reproduction.

(iii) Data quality. The training data is constructed from diverse, model-generated completions with varying degrees of copying, rather than from arbitrary repeated tokens. This exposes the probe to realistic decoding contexts and encourages generalization beyond simple string matching.

(iv) Regularized representation. The probe is a lightweight classifier trained on frozen LLM hidden representations, rather than a module that directly observes the full context, token embeddings, or KV cache.

Finally, the empirical results support this interpretation. Compared with heuristic copy methods such as CopySpec and BanditSpec, *SwitchSD* invokes copying less frequently but achieves longer accepted copy spans and higher throughput. Moreover, when conditioning on positions where a copy candidate exists, probe-active states yield substantially longer accepted spans, whereas probe-inactive states mostly lead to short or zero-token acceptances. Thus, the probe is best understood as learning copy-readiness: an internal signal that predicts when a surface repetition is likely to match the target model's continuation.

F Component Ablation: Routing vs. Multi-Candidate Verification

SwitchSD combines two distinct mechanisms: (i) probe-based routing, which determines when context-based copying is likely to be beneficial, and (ii) parallel verification of multiple retrieved continuations when the copy path is selected. To isolate the contribution of each component, we evaluate a controlled six-arm ablation on the full HumanEval benchmark using LLaMA-3.1-8B-Instruct. All configurations are evaluated on H200 GPUs and report mean $\pm$ sample standard deviation across three runs.

Contribution of Probe-Based Routing. Comparing C2 and C5 isolates the contribution of the latent routing signal while keeping multi-candidate verification enabled. The n-gram trigger with multi-candidate verification already improves over EAGLE3, reaching 128.85 ± 2.11 Tok/s. Adding the probe increases throughput to 132.69 ± 0.55 Tok/s, corresponding to a further 3.0% improvement. Importantly, the probe makes copying more selective: the copy path is invoked on 16.8% rather than 18.5% of speculation iterations, while its mean accepted length increases from 5.06 to 5.48. The probe therefore improves routing primarily by increasing the precision of copy-path activation rather than by increasing its frequency.

Contribution of Multi-Candidate Verification. Comparing C4 and C5 isolates the effect of parallel multi-candidate verification under probe-based routing. Restricting the copy path to a single candidate yields 117.76 ± 3.01 Tok/s, whereas allowing multiple retrieved candidates to be

Experiment	Configuration	Tok/s	vs. EAGLE3	Copy % @ MAL
C0	EAGLE3 (copy off)	122.86 ± 0.80	–	–
C1	CopySpec heuristic	117.87 ± 0.49	-4.1%	$28.1\%@2.32$
C2	n-gram trigger + multi-candidate	128.85 ± 2.11	$+4.9\%$	$18.5\%@5.06$
C3	Probe only (n-gram filter off)	124.29 ± 1.05	$+1.2\%$	$33.2\%@4.05$
C4	Probe + single candidate	117.76 ± 3.01	-4.1%	$26.5\%@2.43$
C5	**Full *SwitchSD***	$\mathbf{132.69 \pm 0.55}$	$\mathbf{+8.0\%}$	$\mathbf{16.8\%@5.48}$

Table 7: **Controlled component isolation on HumanEval with LLaMA-3.1-8B-Instruct.** Throughput is reported as mean $\pm$ sample standard deviation over three runs. MAL denotes the mean accepted length of the copy path. C2 removes the latent probe while retaining multi-candidate verification; C3 disables the structural n-gram filter; C4 retains probe-based routing but restricts copy verification to a single retrieved candidate.

verified in parallel increases throughput to 132.69 ± 0.55 Tok/s, a 12.7% improvement. This result shows that predicting whether copying is useful and identifying the exact historical continuation are distinct problems. Even when the probe correctly detects a copy-oriented regime, committing to a single retrieved continuation is brittle. Retrieving multiple plausible continuations and allowing the target model to resolve this ambiguity within a single block-triangular verification pass substantially increases the yield of context-based drafting.

Probe Signal Beyond the Structural n-gram Rule. The comparison between C2, C3, and C5 tests whether the probe merely reproduces the structural heuristic used in its weak supervision. With the n-gram filter disabled entirely, the probe-only configuration C3 still improves over EAGLE3, reaching 124.29 ± 1.05 Tok/s. Conversely, C2 uses only the n-gram trigger and reaches 128.85 ± 2.11 Tok/s. Combining the two signals in C5 produces the best result, 132.69 ± 0.55 Tok/s. Thus, the latent probe and structural filter provide complementary information.

This distinction is also enforced architecturally. The probe observes only the hidden representation $H_i^{(\ell)}$ at a single decoding position. It does not receive the context token sequence used for n-gram lookup or the retrieved candidate continuations and therefore cannot directly implement string matching. The n-gram condition provides external evidence that a reusable continuation exists in the context, whereas the probe estimates from the model's internal state whether the current decoding regime is compatible with copying.

Complementarity of Routing and Candidate Verification. The full ablation indicates that neither component alone explains the performance of *SwitchSD*. The CopySpec-style heuristic in C1 and probe-based routing with only a single candidate in C4 both underperform EAGLE3 by 4.1%. Multi-candidate verification makes context-based drafting sufficiently robust to be useful, while the latent probe suppresses low-yield invocations of that mechanism. Their combination in C5 achieves both the highest throughput and the highest copy-path mean accepted length in the isolation study.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims regarding the precision of copy-intent probing and the resulting throughput gains are supported by experimental data in Section 4 and the architectural details in Section 3.

   Guidelines: *Standard compliance.*

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations regarding model architectures (Transformers) and deployment settings in Section 6.

   Guidelines: *Standard compliance.*

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A],

   Justification: This paper is primarily an empirical and systems-oriented study; no formal theorems are proposed.

   Guidelines: *NA is appropriate for implementation-focused papers.*

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full details on probe training, dataset construction, and threshold selection in Section 3.2 and Appendix D.

   Guidelines: *Standard compliance.*

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code for SwitchSD and the synthetic data generation scripts are included in the supplementary material zip file.

   Guidelines: *Assuming you are uploading a zip file with the submission.*

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters for the linear probes and the logic for the greedy layer selection are detailed in Section 3.2.

   Guidelines: *Standard compliance.*

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Performance results in Table 1 and Table 2 are reported as means over standard benchmark test sets (HumanEval, Math500, CNN/DM).

Guidelines: *Standard for throughput papers.*

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Answer: [Yes]

   Justification: Hardware specifications (e.g. H100 GPUs) and the inference environment are described in Appendix D.

   Guidelines: *Standard compliance.*

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research on inference optimization does not involve human subjects, sensitive data, or harmful applications.

   Guidelines: *Standard compliance.*

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive impact of reduced computational energy consumption in Appendix F. We find no direct negative societal impacts specific to this optimization.

    Guidelines: *Standard compliance.*

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: SwitchSD is an inference acceleration framework and does not introduce new model weights or high-risk datasets.

    Guidelines: *NA is common for systems papers that do not alter model outputs.*

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All base models (Llama 3, Qwen) and benchmarks (HumanEval) are properly cited and used according to their respective licenses.

    Guidelines: *Standard compliance.*

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new benchmark datasets or large-scale pretrained models.

Guidelines: *The synthetic data is a training aid for the probe, usually not considered a "New Asset" in the benchmark sense.*

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human subjects or crowdsourcing were used in this research.

    Guidelines: *Standard compliance.*

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects were involved in this research.

    Guidelines: *Standard compliance.*

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We explicitly describe the use of Claude Sonnet for generating the synthetic copy-inducing dataset in Section 3.2.

    Guidelines: *A common methodology.*