WhiFlash: Accelerating Speculative Decoding with Token-Level Cross-Paradigm Routing

Young D. Kwon, Miles Williams, Rui Li, Alexandros Kouris, and Stylianos I. Venieris
Samsung AI Center - Cambridge, UK
yd.kwon@samsung.com

Samsung Research
AI Center-Cambridge

Paper Code

TL;DR: The best drafter changes frequently between autoregressive and diffusion drafters, even within a single generation sequence. WhiFlash routes between them dynamically at the token level, significantly improving throughput at minimal overhead.

Introduction

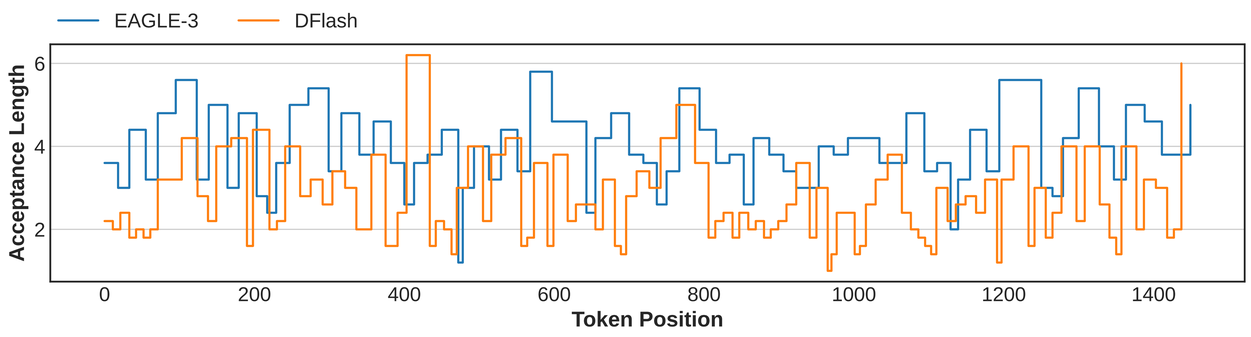
🎨 Autoregressive decoding is the main latency bottleneck of LLM inference. Speculative decoding [1] alleviates it.

✂️ SOTA methods rely on a static drafting paradigm: autoregressive drafters (EAGLE-3 [2]) excel at chat and reasoning, diffusion-based drafters (DFlash [3]) at structured outputs.

🚀 Realistic workloads (e.g. agentic) alternate between reasoning and structured tool calls. Yet existing drafter routing (HedgeSpec [4]) converges on a single drafter per prompt.

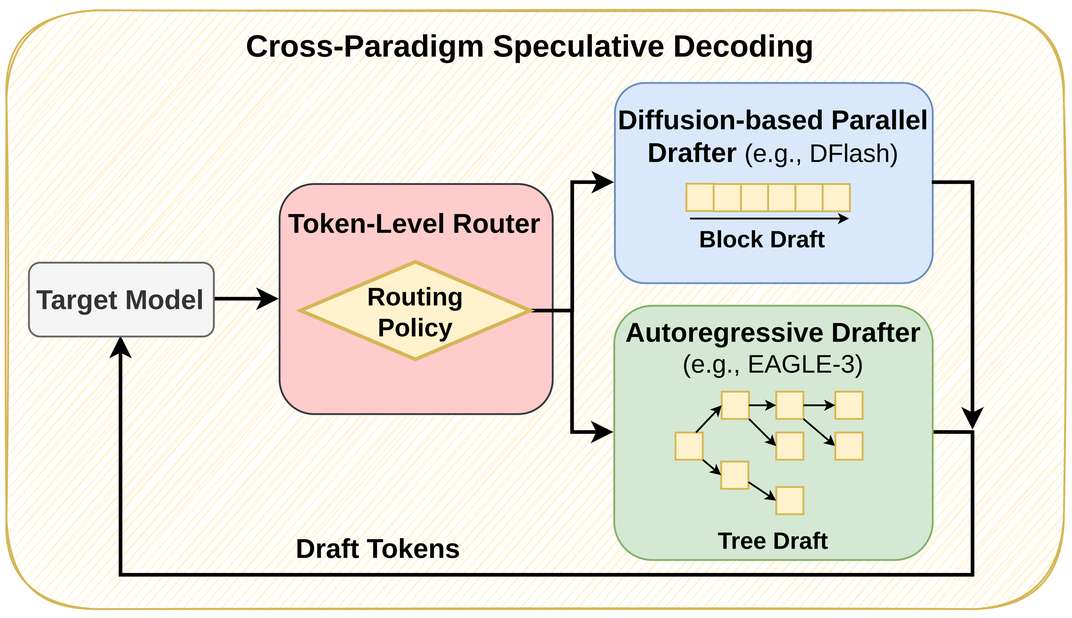
Motivation

🔬 Acceptance length swings between two and six tokens within a single response.



☑ **No paradigm wins a whole sequence,** so task- and prompt-level routing leave gains unrealised.

WhiFlash



A token-level router selects the drafter at every decoding step.

I. **WhiFlash-Entropy:** route using the target model's next-token entropy.
II. **WhiFlash-Neural:** use a lightweight MLP to predict the acceptance-length gains from the target model's hidden states.

System Optimisation

I. **Lazy Catch-up:** update the inactive drafter's KV cache only when needed.
II. **KV-only Prefill:** skip the MLP and LM head for already-accepted tokens.

Together, they keep switching cost below 7% of per-round latency.

Evaluation

- Qwen3-4B, Qwen3-8B and Llama-3.1-8B with EAGLE-3 [2], DFlash [3] and hindsight oracles at task, prompt and token level are used.
- 15 datasets across Chat, Math, Coding and Agentic tasks (measured on NVIDIA H100 and RTX 4090).

Model	Method	Chat		Math		Coding		Agentic		Total Avg.	
		AL	Speedup (×)	AL	Speedup (×)	AL	Speedup (×)	AL	Speedup (×)	AL	Speedup (×)
Qwen3-4B	EAGLE-3	5.21	2.58	5.44	2.94	5.25	2.79	3.99	2.02	4.95	2.57
	DFlash	3.43	2.08	7.22	4.75	6.45	4.29	3.56	2.34	5.08	3.30
	Oracle-Task	5.21	2.58	7.22	4.75	6.45	4.29	3.99	2.34	5.67	3.44
	Oracle-Prompt	5.29	2.76	7.22	5.00	6.49	4.45	4.01	2.41	5.71	3.60
	Oracle-Token	5.68	-	8.36	-	7.79	-	4.81	-	6.59	-
	WhiFlash-Entropy	5.19	2.69	7.36	4.90	6.73	4.46	4.22	2.41	5.82	3.56
	WhiFlash-Neural	**5.41**	**2.78**	**7.66**	**4.98**	**6.97**	**4.57**	**4.40**	**2.42**	**6.05**	**3.63**
Qwen3-8B	EAGLE-3	5.24	2.73	5.58	3.13	5.29	2.90	4.42	2.38	5.12	2.78
	DFlash	3.40	2.12	7.38	5.23	6.56	4.44	4.12	2.62	5.29	3.55
	Oracle-Task	5.24	2.73	7.38	5.23	6.56	4.44	4.45	2.62	5.86	3.71
	Oracle-Prompt	5.34	2.95	7.39	5.20	6.63	4.63	4.53	2.72	5.93	3.82
	Oracle-Token	5.72	-	8.56	-	7.98	-	5.26	-	6.81	-
	WhiFlash-Entropy	5.24	2.84	7.54	5.28	6.97	4.64	4.78	2.73	6.08	3.82
	WhiFlash-Neural	**5.47**	**2.92**	**7.72**	**5.30**	**7.20**	**4.67**	**4.90**	**2.78**	**6.26**	**3.87**

Table 1: Per-category acceptance length (AL) and speedup over vanilla autogressive decoding. Oracle are hindsight upper bounds.

τ_{neural}	Chat	Math	Coding	Agentic
−1.0	149.5	238.4	223.1	136.7
−0.5	147.2	247.7	226.7	137.4
0.0	146.1	249.1	223.7	140.0
0.5	148.1	250.6	228.5	**142.7**
1.0	**150.0**	**263.4**	**237.5**	140.6
1.5	148.4	262.4	235.8	139.4
2.0	143.4	256.2	221.9	134.2

Table 2: Threshold sensitivity analysis w.r.t. throughput (tokens/s) of WhiFlash-Neural (Qwen3-8B).

Prefill Tokens	Backlog Tokens 1	64	128	512	1024
512	3.3%	3.8%	3.8%	3.8%	4.9%
1024	3.3%	3.8%	3.8%	3.9%	5.2%
2048	3.2%	3.8%	3.8%	4.1%	5.6%
4096	2.8%	3.7%	3.7%	4.2%	**5.8%**
8192	1.9%	3.4%	3.4%	3.8%	5.7%

Table 3: Profiling switching cost w.r.t. per-round latency (Qwen3-8B).

Method	Memory (GB)	% Increase
Qwen3-8B (No SD)	15.26	–
EAGLE-3	16.39	7.4%
DFlash	17.62	15.4%
WhiFlash-Entropy/Neural	18.41	20.6%

Table 4: Peak memory of baselines and WhiFlash (Qwen3-8B).

☑ **WhiFlash** achieves up to **69.6% higher throughput over EAGLE-3 & 37.3% over DFlash.**

☑ **Switching cost** stays **below 7%** and **router below 1%** of per-round latency, with **only 5.2% more memory** than DFlash.

☑ **Gains are robust** across a wide threshold range and across hardware (H100 and RTX 4090).

Conclusions

💻 We introduced WhiFlash, the first cross-paradigm speculative decoding method, routing between autoregressive and diffusion-based parallel drafters at the token level.

🏋️ Lazy catch-up and KV-only prefill make per-step switching practical.

References
[1] Leviathan, Y. et al. Fast Inference from Transformers via Speculative Decoding. In International Conference on Machine Learning (ICML'23).
[2] Li, Y. et al. EAGLE-3: Scaling up Inference Acceleration of Large Language Models via Training-Time Test. In Advances in Neural Information Processing Systems (NeurIPS'25).
[3] Chen, J. et al. DFlash: Block Diffusion for Flash Speculative Decoding. In International Conference on Machine Learning (ICML'26).
[4] Liu, H. et al. Not-a-Bandit: Provably No-Regret Drafter Selection in Speculative Decoding for LLMs. In International Conference on Learning Representations (ICLR'26).