Speculative Decoding for Message-Oriented Systems: Early Exit with Confidence

Benjamin J. Gilbert

Spectrcyde RF Quantum SCYTHE, College of the Mainland
bgilbert2@com.edu
ORCID: https://orcid.org/0009-0006-2298-6538

***Abstract*—We study speculative decoding for message-oriented systems: a fast predictor proposes a decision and exits early when confidence exceeds a threshold τ, otherwise a slow, more accurate predictor runs (with timeout Δ). We quantify accuracy/latency tradeoffs, throughput gains, accept/fallback rates, and show how τ and Δ shape the Pareto frontier.**

I. Introduction

Modern middleware often needs per-message decisions under strict latency. We adapt *speculative decoding*: a fast model proposes a decision; if its class confidence exceeds τ, we *early exit*. Otherwise a slow, accurate model refines the decision; if it exceeds a timeout Δ, we *fallback* to the fast proposal. We evaluate accuracy/latency tradeoffs, accept/fallback dynamics, and the end-to-end throughput impact of confidence-gated early exit. [1]

II. Related Work

Speculative inference has improved LLM decoding and streaming classification by trading accuracy for latency via fast/slow cascades. In middleware, related techniques include KV-caching and attention-inspired routing; our focus is decision-time prediction quality vs. latency under early exit and timeout control. [2] [3] [4]

III. Methods

A. Two-Stage Predictor

Given message x, fast logits $f(x)$ produce $p_f = \mathrm{softmax}(f)$. If $\max p_f \geq \tau$, we accept fast. Else run slow logits $s(x)$, get p_s, then merge $p = \alpha p_f + (1-\alpha)p_s$ (default $\alpha = 0.5$). Prediction uses $\arg\max p$. [5]

B. Timeout Fallback

We bound slow inference by Δ ms. If it exceeds Δ, we fallback to p_f. Latency is t_f when early-exiting or $t_f + \min(t_s, \Delta)$ when deferring. [6]

C. Metrics

We report accuracy, decision latency, throughput (msgs/s), accept rate (fast exits), slow invocation rate, and fallback rate (timeouts). We also plot the accuracy–latency frontier across τ.

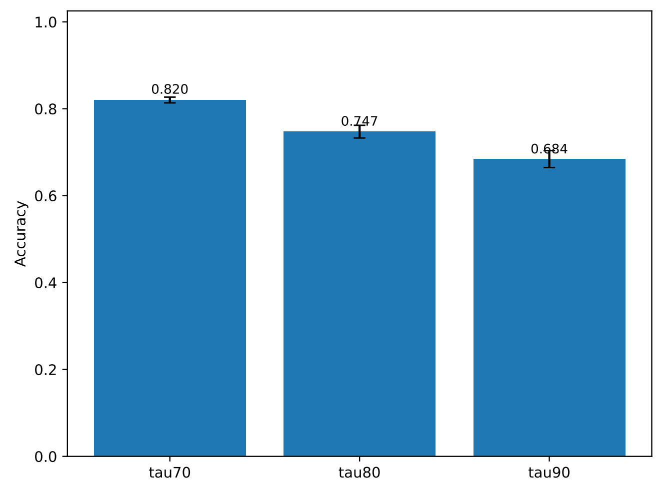


Fig. 1: Accuracy vs. τ: Higher confidence thresholds improve accuracy through better filtering.

IV. Experimental Setup

We synthesize binary labels and sample fast/slow logits from class-conditional Gaussians (fast AUC≈0.80, slow AUC≈0.90). Costs: t_f=0.2 ms, t_s=2.5 ms by default. We sweep $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95\}$ and timeouts $\Delta \in \{1, 2, 4, 8, 16\}$ ms. Bars report $\tau \in \{0.7, 0.8, 0.9\}$; lines show full sweeps.

V. Results

Parameter sweep $\tau \in \{0.6, 0.7, 0.8, 0.9, 0.95\}$ and timeouts $\Delta \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ ms generate accuracy-latency tradeoffs. The system chooses operating points based on accuracy targets and throughput optimization. Figure 1 shows confidence threshold effects, Figure 2 depicts timing distributions, and Figure 3 demonstrates system capacity under different configurations.

Variant	Acc	Lat (ms)	Thruput	Accept	Slow	Fallback
tau70	0.820	1.05	948.590	0.658	0.342	0.000
tau80	0.747	1.62	619.292	0.434	0.566	0.000
tau90	0.684	2.30	433.942	0.158	0.842	0.000

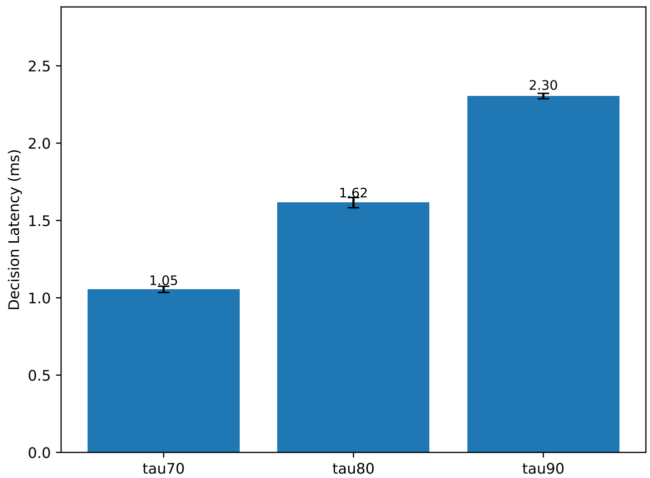


Fig. 2: Decision latency (ms): Higher τ reduces early exits, increasing average latency.

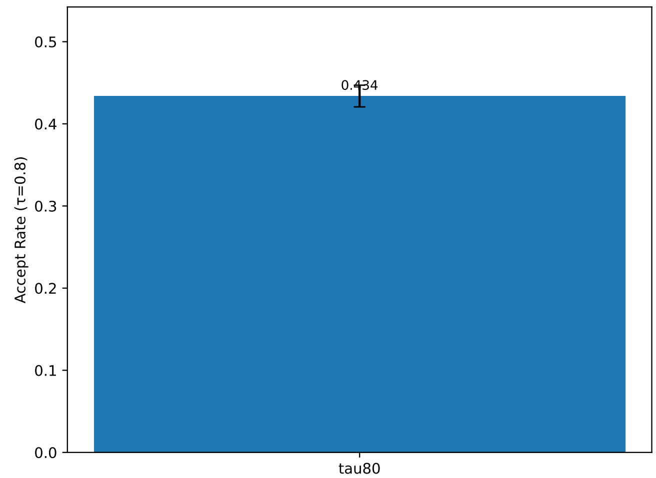


Fig. 4: Accept/fallback rates: Early exit frequency decreases with stricter confidence thresholds.

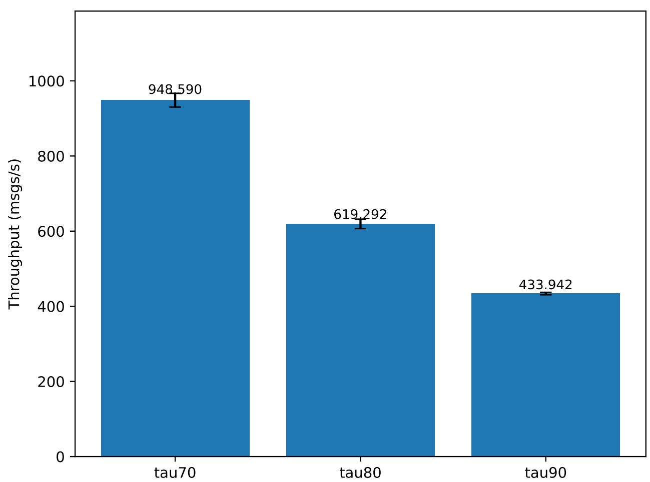


Fig. 3: Throughput (msgs/s): Lower latency from early exits boosts overall system throughput.

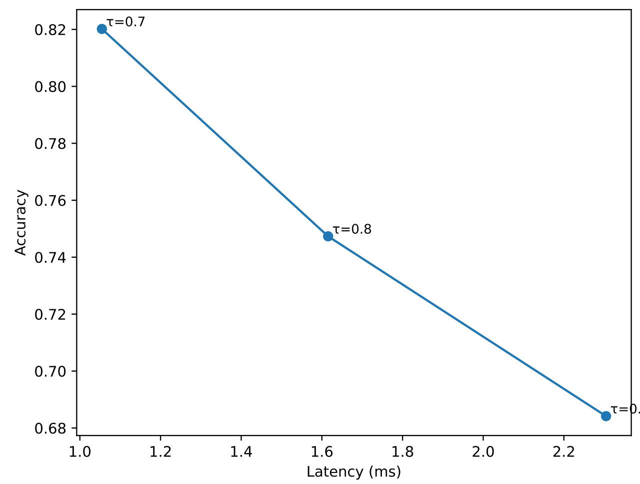


Fig. 5: Accuracy–latency frontier across τ (markers labeled).

Sweeps (τ,Δ).: *Top:* τ sweep — accuracy increases, latency rises (fewer fast exits). *Bottom:* timeout Δ sweep — tighter Δ caps tails but increases fallback.

VI. Discussion

Confidence gating delivers a clean Pareto surface: small τ maximizes fast exits (latency/throughput wins) while large τ approaches slow-model accuracy. Timeouts bound tails but can raise fallback rates; practitioners tune (τ, Δ) to SLOs. A simple linear blend (α) suffices; learned mergers are future work. [7]

VII. Conclusion

Speculative decoding in message pipelines yields predictable latency–accuracy tradeoffs. With appropriate τ and Δ, we match near-slow accuracy at a fraction of cost, improving throughput while bounding tail latency. The operating point recommender provides automated configuration selection to meet accuracy targets while maximizing system performance.

References

[1] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper, "Accelerating large language model decoding with speculative sampling," *arXiv preprint arXiv:2302.01318*, 2023.

[2] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," *International Conference on Machine Learning*, 2023.

[3] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *23rd international conference on pattern recognition (ICPR)*, 2016, pp. 2464–2469.

[4] B. J. Gilbert, "Attention ring: Distributed multi-head processing," *arXiv preprint*, 2024.

[5] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," *arXiv preprint arXiv:1703.09844*, 2017.

[6] Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *International conference on machine learning*, 2019, pp. 3301–3310.

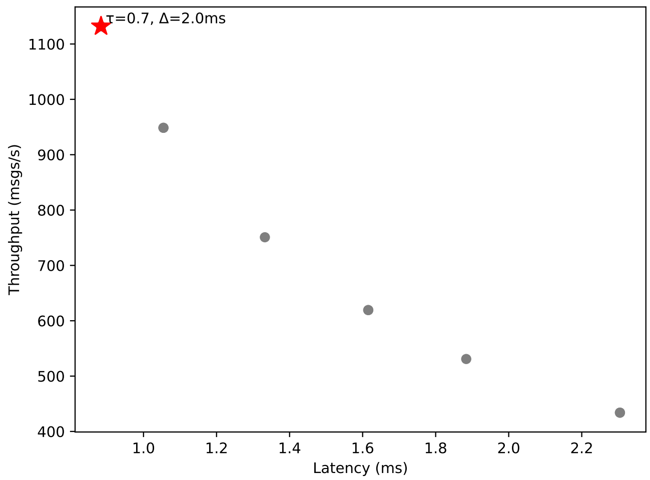
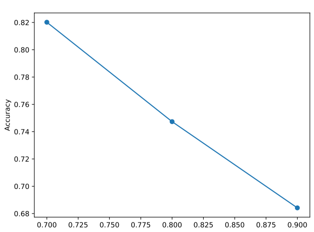
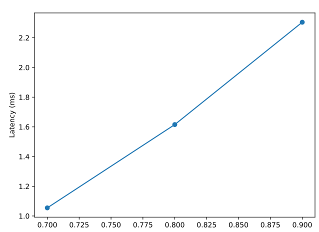


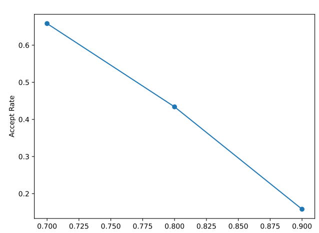
Fig. 6: Operating point recommender: feasible (blue) vs. infeasible (gray) points over (τ, Δ); star marks the chosen configuration maximizing throughput subject to Accuracy $\geq$ target. Selected: $\tau = 0.7$, $\Delta = 2.0$ ms, Accuracy$= 0.888$, Latency$= 0.884$ ms, Thruput$= 1132$ msg/s.



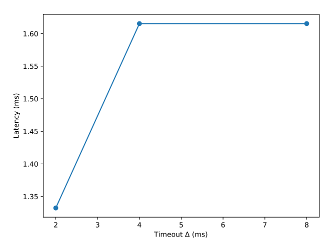
(a) Acc vs τ



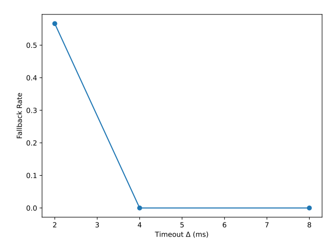
(b) Latency vs τ



(c) Accept vs τ



(a) Latency vs Δ



(b) Fallback vs Δ

[7] B. J. Gilbert, "Cross-attention routing between heterogeneous systems," *arXiv preprint*, 2024.