
Speculative Decoding on One Production Server: 66% of Draft Tokens Accepted in 29 Hours

Yury Akinin (author) · Dmitrii Lozhkin (reviewer) · AVELIN AI · October 20, 2026

Abstract

We report speculative decoding from one production server’s live traffic, which its operator describes as internal agentic sessions: 28.9 hours, 17,056 requests, 4.58 million steps of a multi-token-prediction head drafting three tokens. 66.0% of 13.7 million draft tokens were accepted, between two groups of controlled runs (53.0 to 54.5%, 70.4 to 70.6%); 79.9%, 64.6% and 53.3% of drafts were accepted through the first, second and third token; the main model produced 2.98 tokens per step. Acceptance drifted: of 3,558 ten-second intervals with at least 500 drafts, four of five lay between 56.2% and 75.7%. The mean was flat from one to six or more requests running (64.5 to 65.9%; at most 13 seen). A draft token was accepted about as often as the first whenever the one before it was (79.9%, 80.9%, 82.5%). The geometric formula fed the first-position rate is within 1.1% of measured tokens per step (two of its four terms are exact by construction); fed accepted over drafted tokens, the rate servers report, it is 20% low.

1 Introduction

Speculative decoding lets a cheap draft propose several tokens and the main model check them in one pass, so that the tokens it accepts come almost free [1]. A multi-token-prediction head trained with the model can serve as the draft [2, 4]. The gain is set by how many draft tokens are accepted, and four things are known about that. Acceptance and speed-up depend on the task [1, 3, 4]. The share accepted through position k falls along the draft by construction [4, 5, 8]; the chance given the previous token falls for a head reused without training for it [4] and for a parallel drafter [10], and is flat or rising for an autoregressive drafter [10]. The classic formula for tokens per step assumes independent, equally likely acceptances, which is an approximation [1, 6]. And the speed-up shrinks as the load grows [7, 8]. These results come from benchmark sets, where acceptance already varies between requests [8]. We found no peer-reviewed measurement of acceptance on live traffic. Vendor preprints report speed on live traffic [10] and acceptance length on replayed production traffic [11]; Lookahead (KDD 2024), deployed in Alipay, reports latency of five deployed scenarios and accepted tokens per step on benchmarks [9]. None that we found gives acceptance by interval or draft position from a live server. This note confirms on live traffic that acceptance varies with the workload and adds how it moves between intervals, its relation to load, and a check of the formula.

2 Setup

Production. One NVIDIA H200 runs an open-source inference server with a large mixture-of-experts language model. The draft is the model’s own multi-token-prediction head applied three times. The server counts drafts, draft tokens, accepted tokens, and for each position the drafts accepted through it. That count, divided by the drafts, is the share accepted through position k ; its ratio to the share through $k - 1$ is the chance that draft token k is accepted given that the one before was. Tokens per step is one plus the accepted tokens per draft. The counters cover 28.9 hours, 15.8 in intervals with a draft line: 17,056 finished requests, with a mean prompt of 130,000 tokens and a mean reply of 792, and at most 13 requests running at the end of any interval. Every ten seconds the server writes the same quantities, with the number of requests running; the 5,677 such lines add up to the counters within 0.02%. Where intervals are compared we use the 3,558 with at least 500 drafts: 63% of the lines, 91% of the draft tokens.

Controlled runs. The same model with 8-bit weights ran on another card with the same draft settings at temperature 0.7; counters were read before and after every run. The workloads were a fixed evaluation set (3,582 short requests answered); prompts from public datasets with reasoning on, six of them at one stream and larger sets of up to 192 at 8 to 64 streams; and synthetic prompts, random words with a request for a long essay, cut at 512 tokens. Other clients used that server in some runs: 4% of the generated tokens, 38% in the three six-prompt runs with reasoning on. Runs with under 1% give 53.7% for reasoning on (all nine: 54.8%), 71.4% for synthetic text (70.4%).

The 40-prompt pair. In an earlier start of the production server, with the same settings and before it took traffic, the first 40 English prompts of a fixed public-dataset file (35 code, 5 writing) were answered four at a time, reasoning off, then on, at temperature 0.7, top-p 0.95, at most 512 tokens, ending 15 minutes before operator traffic and 14.6

hours before the counters began. Decoding produced a mean of 393 tokens per request with reasoning on and 375 off. The server’s own lines for the run hold 32,340 draft tokens against 32,319 in the pair, at most 4 requests running.

3 Results

Table 1: Draft acceptance: production, then five controlled runs of the same model. Last column: error of $1 + \alpha + \alpha^2 + \alpha^3$ with α the rate at position 1. † One run per mode on the production server before it took traffic; other rows: 8-bit weights, another card.

Workload	Drafts	Accepted, %	Given the one before, %			Tokens per step	Formula, %
			1	2	3		
Production, 28.9 hours	4,575,261	66.0	79.9	80.9	82.5	2.98	−1.1
Evaluation set, 3,582 requests	86,618	53.0	71.4	71.0	73.0	2.59	−0.1
Larger public-dataset sets, reasoning on	100,416	54.4	72.5	71.7	74.1	2.63	+0.1
40 prompts, reasoning on†	5,969	54.5	73.4	71.1	72.6	2.64	+1.2
The same 40 prompts, reasoning off†	4,804	70.6	84.3	82.8	82.9	3.12	+1.0
Synthetic prompts	76,123	70.4	85.8	79.0	85.2	3.11	+3.6

Two thirds of draft tokens are accepted. The server made 4.58 million drafts and accepted 66.0% of 13.7 million draft tokens (Table 1): 79.9% of drafts through the first token, 64.6% through the second, 53.3% through all three.

Acceptance varies with the workload and drifts in time. In the controlled runs the model accepted 53.0% on the evaluation set, 54.4% on the larger public-dataset sets and 70.4% on synthetic text; on the same 40 prompts, 54.5% with reasoning on, 70.6% off. Reasoning on lowers acceptance length on the same benchmarks for two models [12, 13] and has no fixed sign for a third [13]; ours is one run per mode of a built-in head. The five workloads fall in two groups, 53.0 to 54.5% and 70.4 to 70.6%; production’s 66.0% lies between them. Production intervals vary over a range of that size (Figure 1a). Of the 3,558 intervals, four of five lay between 56.2% and 75.7%; the extremes were 41.5% and 97.8%, the standard deviation 7.6 percentage points (pp). If every draft had the same chances, sampling alone would give 1.3 pp: the variance is 34 times that (32 and 39 times with thresholds of 100 and 1,000 drafts), so the chances differ between intervals. Intervals ten seconds apart correlate at 0.54, thirty seconds at 0.32, five minutes at 0.14 and ten minutes at 0.02: the correlation is gone after about ten minutes.

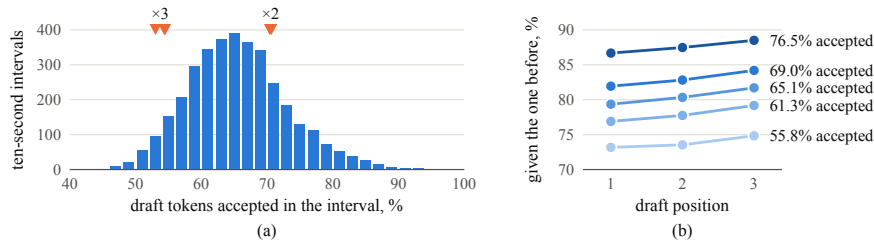


Figure 1: (a) The 3,558 ten-second production intervals with at least 500 drafts, by the share of draft tokens accepted; triangles mark the five controlled workloads of Table 1 (×3 and ×2: marks lying on top of each other). (b) Chance that a draft token is accepted given that the one before it was (at position 1, that it is accepted), for five equal groups of intervals by acceptance.

The mean moves by under 1.5 pp with load in the range we saw. With one, two, three, four, five and six or more requests running at the interval’s end the mean was 64.7%, 65.8%, 65.9%, 65.1%, 65.1% and 64.5%. Intervals that ended with no request running stand apart at 67.7%; they end when the last request finishes. In runs with under 1% other-client tokens, acceptance with prompts from public datasets fell from 59.0% at one stream to 52.7% at 64 (56.6% at 8, 54.9% at 32), but the set of prompts grew with the streams. With synthetic prompts, where it did not, acceptance was 66.4%, 75.8%, 71.1%, 72.2% and 70.8% at 1, 8, 16, 32 and 64 streams, without a trend.

The next draft token is accepted about as often as the first. Given that the first draft token was accepted, the second was accepted in 80.9% of those drafts, and the third, given the second, in 82.5%, against 79.9% for the first. The pattern holds from the fifth of intervals with the lowest acceptance (73.2%, 73.5%, 74.8%) to the highest (86.7%, 87.5%, 88.5%; Figure 1b), which confirms for a built-in head on live traffic what [10] reports for an autoregressive drafter on benchmarks. The classic formula $1 + \alpha + \alpha^2 + \alpha^3$ is therefore close when α is the first-position rate: 2.95 tokens per step against 2.98 measured (−1.1%); where the rates fall along the draft it overestimates [5]. Its first two terms are exact by construction; the last two are 2.8% low, because the later rates are a little higher than the first. On the controlled workloads the error is −0.1% to +1.2%, and +3.6% on synthetic text. With the rate over the tokens that

were checked, accepted over accepted plus rejected (80.9%), the formula gives 2.99 (+0.5%). With accepted over drafted tokens, the rate that servers report, it gives 2.38 (−20%), because the unchecked tokens after a rejection count as misses; [5] finds the same low reading with a draft model (1.7 against 2.7 tokens per round, our arithmetic).

Tokens per step swings by a fifth, the step rate by more. Tokens per step ran from 2.68 at the 10th percentile of intervals to 3.27 at the 90th, a difference of 22% (18 and 26% with thresholds of 1,000 and 100 drafts), and was only weakly related to the step rate (correlations between −0.19 and +0.06 over three thresholds and one to three requests running). The step rate varied 2.3 times between those percentiles at one request running.

4 Limitations

One server, one model family, one draft head of three tokens, and 15.8 hours of traffic with very long prompts: longer drafts, separate draft models and other workloads are not covered, and other draft methods need not share the pattern of Figure 1b; whether it reflects the head’s training for several steps [4] is untested. We did not read the requests, so the traffic type is the operator’s description, and we do not know the clients’ sampling settings; workload means whatever changed between intervals. The running count includes requests still reading their prompt. The band in the abstract depends on the threshold: 56.5 to 80.2% at 100 drafts and 58.0 to 74.5% at 1,000. The five controlled workloads fall in two groups that differ in reasoning mode as well as prompts, so production’s 66.0% between them does not show what the traffic holds. The 40-prompt pair is one run per mode, reasoning off first, with no per-request counters: a direction and a rough size, not an interval. The 512-token cap cut an unknown share of the reasoning, the prompts are short and not the production mix, and the absence of other clients rests on the server’s lines. The other controlled rows differ in prompts, weights, card and output cap. We did not vary any server setting.

5 Recommendations

1. State the workload with every acceptance rate. A benchmark’s acceptance carries that benchmark’s prompts: ours gave 2.59 to 3.12 tokens per step, production 2.98, and the middle four fifths of its intervals 2.68 to 3.27.
2. Read tokens per step from the server’s own mean, one plus the accepted tokens per draft. If only rates are published, put the first-position rate or the rate over checked tokens into the geometric formula, never accepted over drafted tokens. We did not test the formula at other draft lengths.
3. Compare two builds on the same prompts and report the output length.

Disclosure. The analysis was produced with AI coding agents under the author’s direction. Numbers come from the server’s counters and lines and the controlled runs’ counters; the pair is the run script’s own report, within 0.07% of the server’s lines.

References

- [1] Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023. arXiv:2211.17192.
- [2] F. Gloeckle, B. Y. Idrissi, B. Rozière, et al. Better & faster large language models via multi-token prediction. In *ICML*, 2024. arXiv:2404.19737.
- [3] H. Xia, Z. Yang, Q. Dong, et al. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of ACL*, 2024. arXiv:2401.07851.
- [4] Y. Cai, X. Liang, X. Wang, et al. FastMTP: Accelerating LLM inference with enhanced multi-token prediction. arXiv:2509.18362, 2025.
- [5] P. Chordiya. Lossless but not free: An empirical anatomy of speculative decoding on consumer hardware. arXiv:2607.17283, 2026.
- [6] J. Mamou, O. Pereg, D. Korat, et al. Dynamic speculation lookahead accelerates speculative decoding of large language models. In *NeurIPS ENLSP Workshop*, 2024. arXiv:2405.04304.
- [7] X. Liu, C. Daniel, L. Hu, et al. Optimizing speculative decoding for serving large language models using goodput. arXiv:2406.14066v2, 2024 (v3, 2025, is titled TurboSpec and has other authors).
- [8] X. Liu, J. Yu, J. Park, et al. Speculative decoding: Performance or illusion? In *MLSys*, 2026. arXiv:2601.11580.
- [9] Y. Zhao, Z. Xie, C. Liang, et al. Lookahead: An inference acceleration framework for large language model with lossless generation accuracy. In *KDD*, 2024. arXiv:2312.12728.
- [10] X. Cheng, X. Yu, C. Shao, et al. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. arXiv:2607.05147, 2026.
- [11] H. Liu, R. Cen, J. Shi, et al. AngelSpec: Towards real-world high performance inference with speculative decoding. arXiv:2607.25852, 2026.
- [12] H. Lei, Y. Li, H. Zhang, et al. Draft-OPD: On-policy distillation for speculative draft models. arXiv:2605.29343, 2026.
- [13] D. Eldenk, P. Mohapatra, Y. Comlek, et al. Attention Drift: What autoregressive speculative decoding models learn. arXiv:2605.09992, 2026.