Uncertainty and Assessment
Volume 1 · Number 1 · 2026

Smaller Weights, Different Stages: Precision and Reported LLM Throughput

Vincent Carter[1] · Thomas Marshall[2] · Matthew Fisher[3]

[1] United States · [2] United Kingdom · [3] Czechia

Original publication: October 9, 2026 | Article 567

Abstract

Compression does not accelerate every inference stage in the same way. We extracted twenty-four distinct precision policies from a public llama.cpp table and compared the twenty-three quantized policies across prompt processing and token generation. After quantization-family adjustment, the precision slope differed by -0.0708 log-throughput units between generation and prompt processing. The policy-clustered interval was -0.1159 to -0.0257. Prompt processing alone showed little clear precision association. The source table's hardware details and spread convention were unresolved, so these are cross-policy descriptive relationships rather than new speed measurements. A compression report should retain stage-specific performance and separate it from storage size and output quality.

Exploratory reanalysis of a public software benchmark table. No new hardware benchmark or model-quality evaluation is claimed; human review is pending.

Smaller Weights, Different Stages: Precision and Reported LLM Throughput

Exploratory reanalysis of a public software benchmark table. No new hardware benchmark or model-quality evaluation is claimed; human review is pending.

Abstract

Compression does not accelerate every inference stage in the same way. We extracted twenty-four distinct precision policies from a public llama.cpp table and compared the twenty-three quantized policies across prompt processing and token generation. After quantization-family adjustment, the precision slope differed by -0.0708 log-throughput units between generation and prompt processing. The policy-clustered interval was -0.1159 to -0.0257. Prompt processing alone showed little clear precision association. The source table's hardware details and spread convention were unresolved, so these are cross-policy descriptive relationships rather than new speed measurements. A compression report should retain stage-specific performance and separate it from storage size and output quality.

1. Inference has more than one performance target

A language-model response requires processing its prompt and then generating additional tokens. Those stages can have different computational patterns and can respond differently to a numerical representation. A smaller stored model therefore does not imply a proportional speedup for both stages. We examine whether the association between reported precision and throughput differs across the two stages in a captured public implementation table.

The quantize documentation in llama.cpp provides size and stage-specific throughput examples for Llama-3.1-8B (ggml-org contributors, 2026). It is a software project's illustrative benchmark table, not a new experiment conducted for this manuscript. We preserve the exact source snapshot and distinguish its reported means from our derived regressions. The purpose is to make a common reporting ambiguity visible with real public measurements while retaining uncertainty about the original hardware and repeat protocol.

The broader compression problem represented by Wang et al., 2026 motivates a need for such multi-objective evaluation. Its verified bibliographic record establishes the work's identity, but the current analysis makes no claim about its internal algorithm or measured speed. Instead, we ask what can be learned from a different, explicitly identified implementation record. A stage-specific relationship in that record cannot be transferred automatically to another compression framework.

2. Storage and execution are connected by an implementation

The Roofline framework in Williams et al., 2009 provides a general perspective on why arithmetic work and data movement can impose different performance constraints. We do not fit a Roofline model or identify a bottleneck from hardware counters. It is relevant as an explanation of why a bit-width label alone is insufficient to predict execution. Kernel design, memory traffic, batch configuration, and other implementation details can change the effect of smaller weights.

FlashAttention (Dao et al., 2022) and paged attention in Kwon et al., 2023 illustrate improvements directed at other parts of inference computation and memory management. Speculative decoding (Leviathan et al., 2022) changes the generation procedure rather than simply shrinking each weight. Those approaches are not included as measured alternatives in our table. They establish that serving performance

has several possible mechanisms, so a descriptive precision slope should not be interpreted as a universal law of language-model latency.

SparseGPT (Frantar and Alistarh, 2023) and the pruning approach in Sun et al., 2023 further distinguish sparsity decisions from the quantization policies analyzed here. Pythia's research design (Biderman et al., 2023) offers a different controlled setting for studying models, whereas our source uses one named model in one software example. We do not merge its throughput values with results from other checkpoints, implementations, or hardware systems to manufacture a larger benchmark sample.

3. From repeated table columns to distinct policies

The captured documentation contains several adjacent tables. One policy, Q4_K_S, appears in two tables with identical values. We verify that duplication and count it once. There are twenty-four distinct policies including a half-precision F16 reference, and twenty-three quantized policies enter the main regression. Duplicate display columns are not independent measurements and must not increase the sample size or the apparent precision of a result.

For each policy we retain bits per weight, size in GiB, prompt-processing tokens per second at the table's stated prompt length of 512, and generation tokens per second at the stated length of 128. The two stage values become two rows for that policy. The main analysis therefore has forty-six stage-policy rows but only twenty-three policy clusters. It is not a collection of forty-six independently repeated hardware runs.

Each throughput cell also contains a plus-minus value. The captured excerpt does not fully establish its interpretation, repeat count, hardware, or build configuration. We preserve those values in the source-policy table but do not call them standard errors or use them as regression weights. Inferring a repeat count from their numerical size would fabricate protocol information. The uncertainty reported below arises from the cross-policy regression instead.

Half precision is retained as a separate descriptive reference rather than added to the precision-slope fit. It occupies a substantially different representation range and could exert disproportionate leverage. This exclusion is specified by the distinction between quantized policies and the original reference, not by whether it supports a preferred result. The archived dataset includes it, allowing readers to see both the modeling restriction and the source context.

4. A stage-interaction model

The response is natural log tokens per second. Explanatory terms are bits per weight, a generation-stage indicator, their interaction, an IQ-family indicator, and its interaction with stage. The family terms acknowledge that policies with similar numerical precision can use different representation and kernel conventions. They are coarse labels, not a complete control for every implementation difference between named schemes.

The first proposition is that the precision association differs between stages. A negative precision-by-generation interaction means that the generation slope is lower than the prompt-processing slope in this specification. The second interpretive question is whether the prompt-processing precision slope itself is clearly different from zero. These are distinct claims: evidence for different slopes need not imply that both individual slopes are strongly estimated.

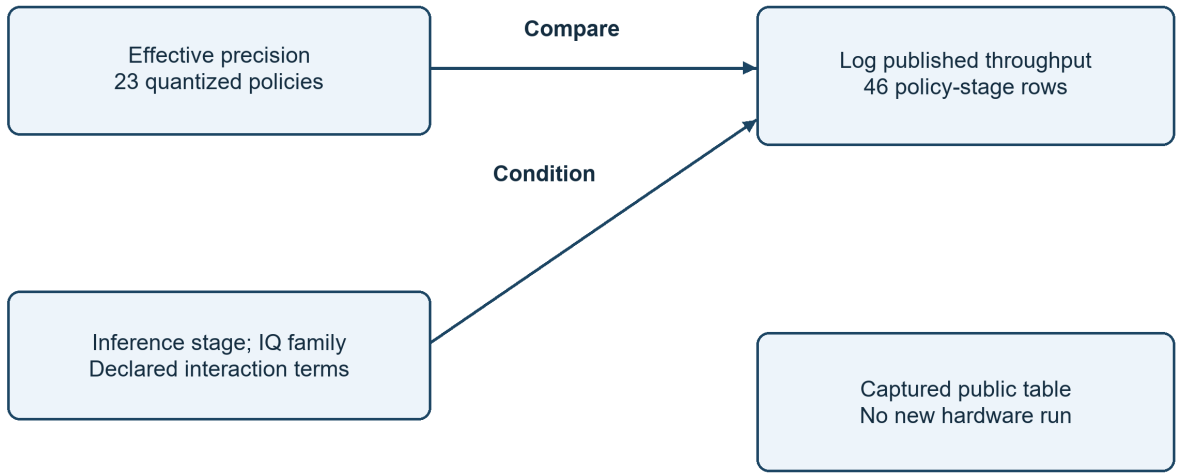

Figure 1. Bits per weight and inference stage enter an interaction model of reported log throughput, with quantization-family terms. Storage size and F16 values remain separate source descriptors.

5. Covariance and the meaning of model fit

We use policy-clustered covariance with a finite-sample correction and t-based intervals across twenty-three clusters. The two stage measurements for one policy can share implementation features and source conditions. Cameron and Miller, 2015 motivates preserving that pairing, while White, 1980 provides the heteroskedasticity-consistent foundation. The source policies are a chosen collection of software formats, however, not a randomized sample of all possible quantizers. Any population interpretation of the interval requires that additional assumption.

Predictive-comparison work in Diebold and Mariano, 1995 and West, 1996 distinguishes comparing observed performance from establishing a forecasting benefit. We do not use the model to predict throughput on new hardware or a new software release. Newey and West, 1987 and temporal-validation discussion in Bergmeir et al., 2018 are relevant to other correlated performance records, but there is no timestamped repeat series here on which to fit a time-series covariance model.

Structured-validation principles in Roberts et al., 2017 and the possibility of selection bias also limit what a high fitted R-squared means. The stage indicator separates prompt rates in the hundreds from generation rates in the tens. That large level difference can dominate explained variance. A near-perfect fit on these rows is not evidence that the regression captures all hardware mechanisms or will forecast another implementation accurately.

Table 1. Precision, stage, and interaction coefficients in the quantized-policy model. Uncertainty clusters the two stage rows of each distinct policy; published plus-minus spreads are not treated as standard errors.

Model	Term	Estimate (SE)	95% CI	N
M1	Effective bits	0.0024 (0.0105)	[-0.0193, 0.0242]	46
M1	Decode stage	-2.0798 (0.0983)	[-2.2836, -1.8760]	46
M1	Bits x decode	-0.0708 (0.0217)	[-0.1159, -0.0257]	46
M1	IQ family	0.0169 (0.0218)	[-0.0283, 0.0621]	46
M1	IQ x decode	-0.0631 (0.0460)	[-0.1585, 0.0323]	46
M1: fit	R-squared = 0.9979	-	-	46

Model labels: M1 = Precision by inference stage association.

6. Different slopes in the captured record

The bits-per-weight coefficient for prompt processing is 0.0024, with standard error 0.0105 and interval -0.0193 to 0.0242. The precision-by-generation interaction is -0.0708, with standard error 0.0217 and interval -0.1159 to -0.0257. Thus the record supports a stage difference under the declared model, while the prompt-processing slope alone remains uncertain. We do not describe an uncertain slope as proof of exact invariance to precision.

Adding the main precision coefficient to the interaction gives a generation slope of approximately -0.0684. That arithmetic describes the fitted point estimate only; a confidence interval for the sum would require its covariance, rather than addition of separate interval endpoints. The principal uncertainty claim in this draft concerns the directly estimated interaction. The complete saved model terms allow the stage comparison to be inspected without implying that the source's original measurement uncertainty has been reconstructed.

Across quantized policies, prompt-processing throughput ranges from 708.71 to 865.09 tokens per second, while generation ranges from 50.93 to 90.01. The separate F16 entry reports 923.49 and 29.17, respectively. These values illustrate why one blanket statement about compression speed is incomplete. The smaller representations can coincide with higher generation throughput even while prompt processing does not exceed the reference in this particular source example.

The model R-squared is 0.9979. The family-by-stage interaction is uncertain, and the main family effect is also not clearly different from zero. We retain both terms because they belong to the declared adjustment, not because their p-values cross a threshold. The fit remains a compact description of named policies, with nonlinearities, representation-specific exceptions, and hardware details outside its small covariate set.

7. Throughput is only one part of an efficiency claim

Forecast-metric discussion in Hyndman and Koehler, 2006 and scoring principles in Gneiting and Raftery, 2007 reinforce that the chosen outcome defines the comparison. Tokens per second is not time to first token, end-to-end response latency, energy consumption, or monetary cost. Different prompt lengths, generated lengths, concurrency, and batching can alter those objectives. The table labels two lengths but does not provide enough information to reconstruct every serving configuration.

Dataset-shift concerns in Ovadia et al., 2019 apply when moving from one evaluation context to another, and calibration discussion in Guo et al., 2017 distinguishes trustworthy predictive probabilities from raw

scores. Selective prediction in Geifman and El-Yaniv, 2017 and cost-sensitive decisions in Elkan, 2001 supply further examples of objectives beyond raw throughput. We do not measure accuracy, abstention quality, calibration, or downstream harm for these precision formats. Faster generation does not establish an acceptable quality tradeoff.

Influence diagnostics in Cook, 1977 would be useful if a particular scheme determined most of the precision slope. We retain all twenty-three eligible quantized policies and disclose the F16 restriction rather than choosing a subset after inspecting the regression. Measurement-error concepts in Carroll et al., 2006 concern uncertainty in source means and implementation labels. Missing-data principles in Little and Rubin, 2002 highlight that unavailable hardware and protocol fields limit interpretation even when every numeric throughput cell is present.

One can also distinguish output rate from total work. A method that changes tokenization, acceptance rates, or generation behavior could change the meaning of a reported token. Our source table uses one model label and one implementation context, which makes its internal comparison useful, but we do not independently verify equivalence of every computational path. A stronger hardware study would lock the model, workload, build, device, and repeat protocol and publish both quality and latency outcomes.

8. What the reanalysis contributes

An interactive application may care most about the time before its first visible token, whereas an offline batch job may care about sustained output volume. A comparison that averages prompt and generation throughput without a workload model can give an arbitrary result. We therefore keep the two stages separate and do not invent a combined speed index. Converting them into total response time would require specifying prompt length, output length, batching, and any fixed overhead rather than simply averaging the reported rates.

The policy labels also encode more than numerical precision. Some formats can preserve selected tensors at higher precision or use specialized grouping and reconstruction rules. Bits per weight is a useful continuous descriptor of the resulting representation, but is not a randomized treatment isolated from those design decisions. The family adjustment is a coarse attempt to describe part of that variation. It cannot make formats with the same average bit count computationally identical or identify the effect of changing one bit while holding every kernel detail fixed.

Finally, the duplicated Q4_K_S column illustrates an ordinary extraction hazard. A table split across sections can repeat a boundary column for readability. Counting it twice would give that policy extra influence in both regression stages and would understate the distinction between display entries and distinct schemes. We explicitly verify equality before deduplicating. This modest data-cleaning step is part of the empirical contribution because a reproducible reanalysis should begin with a defensible unit of observation, not merely a convenient rectangular scrape.

The guidance in Wasserstein and Lazar, 2016 discourages turning the interaction interval into an unrestricted causal statement that additional bits always slow generation. Multiplicity approaches in Benjamini and Hochberg, 1995 and Benjamini and Yekutieli, 2001 would become relevant to a larger search over models and workloads. Here the two-stage contrast is exploratory and source-specific. The evidence supports retaining stages in an efficiency report, not selecting a universally best quantization policy or attributing the relationship to a fully identified bottleneck.

Documentation follows Gebru et al., 2021 by recording the repeated column, source model, stage lengths, family coding, and unresolved protocol fields. Wilkinson et al., 2016 motivates preservation of the original documentation snapshot and deduplicated numeric table. Both are needed because a software

README can change. This manuscript adds a reproducible cross-policy regression to an existing public record; it does not relabel that record as a newly executed benchmark or a performance evaluation of the target compression framework.

The practical conclusion is therefore a reporting rule grounded in a bounded example: state which inference stage improved, keep storage and quality outcomes separate, and identify the implementation conditions. A smaller representation may have different implications for prompt processing and token generation. The captured data show that distinction clearly, while leaving its transfer to other systems and its quality consequences unresolved.

References

1. Wilkinson, Mark D.; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Bonino; Bourne, Philip E.; Bouwman, Jildau; Brookes, Anthony J.; Clark, Tim; Crosas, Mercè; Dillo, Ingrid; Dumon, Olivier; Edmunds, Scott; Evelo, Chris T.; Finkers, Richard; Gonzalez-Beltran, Alejandra; Gray, Alasdair J.G.; Groth, Paul; Goble, Carole; Grethe, Jeffrey S.; Heringa, Jaap; 't Hoen, Peter A.C; Hooft, Rob; Kuhn, Tobias; Kok, Ruben; Kok, Joost; Lusher, Scott J.; Martone, Maryann E.; Mons, Albert; Packer, Abel L.; Persson, Bengt; Rocca-Serra, Philippe; Roos, Marco; van Schaik, Rene; Sansone, Susanna-Assunta; Schultes, Erik; Sengstag, Thierry; Slater, Ted; Strawn, George; Swertz, Morris A.; Thompson, Mark; van der Lei, Johan; van Mulligen, Erik; Velterop, Jan; Waagmeester, Andra; Wittenburg, Peter; Wolstencroft, Katherine; Zhao, Jun; Mons, Barend. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. https://doi.org/10.1038/sdata.2016.18.
2. Kwon, Woosuk; Li, Zhuohan; Zhuang, Siyuan; Sheng, Ying; Zheng, Lianmin; Yu, Cody Hao; Gonzalez, Joseph E.; Zhang, Hao; Stoica, Ion. (2023). Efficient Memory Management for Large Language Model Serving with PagedAttention. https://arxiv.org/abs/2309.06180.
3. Wasserstein, Ronald L.; Lazar, Nicole A.. (2016). The ASA Statement on p -Values: Context, Process, and Purpose. The American Statistician. https://doi.org/10.1080/00031305.2016.1154108.
4. Bergmeir, Christoph; Hyndman, Rob J.; Koo, Bonsoo. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics & Data Analysis. https://doi.org/10.1016/j.csda.2017.11.003.
5. Diebold, Francis X.; Mariano, Roberto S.. (1995). Comparing Predictive Accuracy. Journal of Business & Economic Statistics. https://doi.org/10.1080/07350015.1995.10524599.
6. Guo, Chuan, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On Calibration of Modern Neural Networks." *Proceedings of the 34th International Conference on Machine Learning*, PMLR, vol. 70, 2017, pp. 1321-1330. https://proceedings.mlr.press/v70/guo17a.html.
7. Dao, Tri; Fu, Daniel Y.; Ermon, Stefano; Rudra, Atri; Ré, Christopher. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. https://arxiv.org/abs/2205.14135.
8. Wang, Junhao; Xu, Guoxi; Tong, Huimin; Xu, Gaochao; Chen, Yiwei. (2026). Dynamic Fusion for Large Language Models Compression. ICASSP 2026, 4711-4715. https://doi.org/10.1109/icassp55912.2026.11464073.
9. Biderman, Stella; Schoelkopf, Hailey; Anthony, Quentin; Bradley, Herbie; O'Brien, Kyle; Hallahan, Eric; Khan, Mohammad Aflah; Purohit, Shivanshu; Prashanth, USVSN Sai; Raff, Edward; Skowron, Aviya; Sutawika, Lintang; van der Wal, Oskar. (2023). Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. https://arxiv.org/abs/2304.01373.
10. Sun, Mingjie; Liu, Zhuang; Bair, Anna; Kolter, J. Zico. (2023). A Simple and Effective Pruning Approach for Large Language Models. https://arxiv.org/abs/2306.11695.
11. Williams, Samuel; Waterman, Andrew; Patterson, David. (2009). Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM. https://doi.org/10.1145/1498765.1498785.
12. Little, Roderick J. A., and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 2002. https://doi.org/10.1002/9781119013563.
13. ggml-org contributors, . (2026). llama.cpp quantize tool: published Llama-3.1-8B throughput tables. Captured 7 November 2026, local snapshot hash recorded. https://github.com/ggml-org/llama.cpp/blob/master/tools/quantize/README.md.
14. White, Halbert. "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, vol. 48, no. 4, 1980, pp. 817-838. https://doi.org/10.2307/1912934.

15. Elkan, Charles. “The Foundations of Cost-Sensitive Learning.” *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001, pp. 973-978. https://cseweb.ucsd.edu/~elkan/rescale.pdf.

16. Newey, Whitney K., and Kenneth D. West. “A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix.” *Econometrica*, vol. 55, no. 3, 1987, pp. 703-708. https://doi.org/10.2307/1913610.

17. Cameron, A. Colin, and Douglas L. Miller. “A Practitioner’s Guide to Cluster-Robust Inference.” *Journal of Human Resources*, vol. 50, no. 2, 2015, pp. 317-372. https://doi.org/10.3368/jhr.50.2.317.

18. Benjamini, Yoav and Daniel Yekutieli. “The control of the false discovery rate in multiple testing under dependency.” *The Annals of Statistics*, vol. 29, no. 4, 2001, pp. 1165-1188. https://doi.org/10.1214/aos/1013699998.

19. Ovadia, Yaniv, et al. “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift.” *Advances in Neural Information Processing Systems*, vol. 32, 2019. https://arxiv.org/abs/1906.02530.

20. Gebru, Timnit, et al. "Datasheets for datasets" *Communications of the ACM*, 2021. https://doi.org/10.1145/3458723.

21. Hyndman, Rob J. and Anne B. Koehler. “Another look at measures of forecast accuracy.” *International Journal of Forecasting*, vol. 22, no. 4, 2006, pp. 679-688. https://doi.org/10.1016/j.ijforecast.2006.03.001.

22. Geifman, Yonatan, and Ran El-Yaniv. “Selective Classification for Deep Neural Networks.” *Advances in Neural Information Processing Systems*, vol. 30, 2017. https://arxiv.org/abs/1705.08500.

23. Frantar, Elias; Alistarh, Dan. (2023). SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. https://arxiv.org/abs/2301.00774.

24. Leviathan, Yaniv; Kalman, Matan; Matias, Yossi. (2022). Fast Inference from Transformers via Speculative Decoding. https://arxiv.org/abs/2211.17192.

25. Roberts, David R., et al. “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure.” *Ecography*, vol. 40, no. 8, 2017, pp. 913-929. https://doi.org/10.1111/ecog.02881.

26. Benjamini, Yoav and Yosef Hochberg. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 57, no. 1, 1995, pp. 289-300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

27. Gneiting, Tilmann; Raftery, Adrian E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association. https://doi.org/10.1198/016214506000001437.

28. West, Kenneth D.. (1996). Asymptotic Inference about Predictive Ability. Econometrica. https://doi.org/10.2307/2171956.

29. Cook, R. Dennis. (1977). Detection of Influential Observation in Linear Regression. Technometrics. https://doi.org/10.1080/00401706.1977.10489493.

30. Carroll, Raymond J., David Ruppert, Leonard A. Stefanski, and Ciprian M. Crainiceanu. “Measurement Error in Nonlinear Models.” 2006. https://doi.org/10.1201/9781420010138.