TEMPQUANT: LEARNING ADAPTIVE TEMPERATURES FOR QUANTIZED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly deployed at low precision for applications ranging from reasoning to creative writing and beyond, yet quantization can degrade their performance. These applications often use different decoding settings, including sampling temperature and top-p, which control the randomness of token selection. We study the interaction between quantization and these settings by examining how quantization distorts output token probabilities. These distortions can change generated responses and affect task performance. We characterize the connection between quantization and temperature by measuring changes in generation-time entropy across deployed temperatures, and examine whether these changes can be understood as a temperature-like shift. We also investigate how much distributional error can be corrected through our intervention method, TempQuant. TempQuant is a compact learned predictor that adaptively sets the quantized model's temperature and top-p value at each decoding step to minimize divergence between the full-precision bfloat16 (BF16) and quantized probability distributions. At deployment, it uses only features of the quantized model's outputs, requiring no full-precision reference. We show that TempQuant can improve model accuracy and analyze how these corrections affect model behavior during generation.

1 INTRODUCTION

At every step of autoregressive generation, large language models generate a probability distribution over the next token. This probability distribution is a function of the prefix, which is the prompt together with any tokens generated so far. Based on the chosen decoding method, we select a token from this distribution (Shi et al., 2024). When sampling, the randomness of token selection is controlled by the temperature τ, a user-chosen deployment setting. The chosen temperature depends on the use case: higher temperatures generally lead to more diverse outputs, whereas lower temperatures concentrate sampling on the most likely tokens. Note that changing the temperature affects sampling by changing the entropy of the token probability distribution (how spread out it is).

Quantization is a separate form of compression, where we use fewer bits to represent the weights and, in some configurations, activations of the large language model, so it can fit in less memory and run faster on supporting hardware (Frantar et al., 2023; Liu et al., 2023). Compressing this information introduces numerical error, so the next-token probability distributions can diverge from those of the full-precision model. This noise can also change their entropy in an unintended way.

We aim to analyze how quantization changes the entropy of token probability distributions, whether the change is systematic, and how it connects to the temperature you deploy your model at. If it is systematic, can part of this change be understood as a temperature shift? And can we correct it by calibrating the temperature per token? We measure how much distributional error temperature correction can remove. On shared prefixes, the direction of the mean entropy change depends on which model generated the histories. In the provisional own-generation sweep, MXFP4 has slightly higher mean entropy at all three deployed temperatures. The sparse temperature grid does not resolve an entropy-equivalent BF16 temperature.

We quantify the difference between the full-precision and quantized probability distributions through Kullback–Leibler divergence (Kullback & Leibler, 1951). We hypothesize that reducing this diver-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

gence can help preserve the full-precision model's behavior. This is motivated by *Accuracy is Not All You Need*, in which Dutta et al. (2024) argue that quantization should preserve the full-precision model's behavior, and that accuracy alone does not reflect this because correct-to-incorrect and incorrect-to-correct "flips" can cancel out. They motivate distance metrics such as KL divergence to capture changes that aggregate accuracy hides, so we aim to intervene here.

Minimizing divergence between probability distributions is not new: LLM-QAT updates quantized model weights to minimize forward KL for fixed teacher targets (Liu et al., 2023). However, quantization-aware training (QAT) requires retraining the language-model weights, which can be expensive. We want to reduce this distributional divergence through a lightweight method applicable after post-training quantization (PTQ). We change two settings per token, temperature and top-p, rather than the model weights.

Naturally, this approach is limited compared to methods that update the weights since temperature scaling cannot change token rankings at a fixed prefix, and top-p filtering cannot reorder the tokens it retains. This is a trade-off we make to avoid retraining the language model. It also allows us to ask an interesting question: how much of the degradation due to quantization can be corrected through temperature rescaling? We investigate such questions to see if quantization error introduces a systematic shift in model temperature. This question is also motivated by observations of repetition under greedy decoding (Holtzman et al., 2020) and repetition in low-bit quantized models (Li et al., 2024). A plausible hypothesis is that quantization noise causes a systematic change in model temperature that can contribute to repetition.

For some intuition on how decoding settings change token probabilities, we provide an illustrative example in Figure 1. If we have both distributions, we can find a teacher-informed optimal temperature for the unfiltered quantized distribution within a bounded range (Appendix E). With top-p as a second control, we can jointly optimize temperature and the cutoff to minimize KL divergence. The tricky part is that we do not have the full-precision distribution at inference, and therefore have to approximate these settings from quantized model features. We learn the correction from 42 features of the quantized model's current logits: 32 rank-ordered logit features and ten distribution summaries.

Our approach of reducing divergence between the quantized and full-precision models is motivated by the goal of reproducing the full-precision probability distribution as closely as possible under the chosen reference decoding setting. We measure this divergence using forward KL, JSD, and reverse KL, and test whether reducing these divergences also improves downstream generation, rather than assuming that lower divergence guarantees better accuracy. Reproducing the reference model's behavior differs from maximizing accuracy. This is why, alongside standard accuracy benchmarks, our evaluation includes total variation, similarity to full-precision-generated responses, correctness disagreement, lexical diversity, repetition, and response length.

2 Related Work

Adaptive decoding. Adaptive temperature scaling (ATS) and TempNet also learn token-level temperatures (Xie et al., 2024; Qiu et al., 2024), but they do so for different reasons. ATS learns token-level temperatures to improve confidence calibration of the model, and TempNet learns token-level temperatures to improve prediction and generalization across tasks. ReSET also sets the temperature per token to improve model performance, targeting NVFP4-quantized reasoning models. It calibrates a low temperature for each model/task pair on held-out data, then uses contextual entropy to select between preset temperatures (Lee et al., 2026). It does not learn a temperature predictor, though, and its motivation is to balance precision and exploration during reasoning, rather than imitate full precision. AutoDeco is the closest work to us architecturally, as they also learn temperature and top-p at each generation step. They use cross-entropy to increase the probability of the next token in reasoning traces selected for correct final answers (Wang et al., 2025). TempQuant instead learns these settings to make the quantized model's next-token probabilities match those of its full-precision counterpart.

Temperature calibration. Other papers have tried to learn temperatures for different reasons, for example to calibrate the model. DACA fits temperature with reference-based forward KL on examples where the model and reference predictions agree (Luo et al., 2025). Thermometer predicts one temperature for a task by aggregating unlabeled prompts (Shen et al., 2024). Dual-Align combines

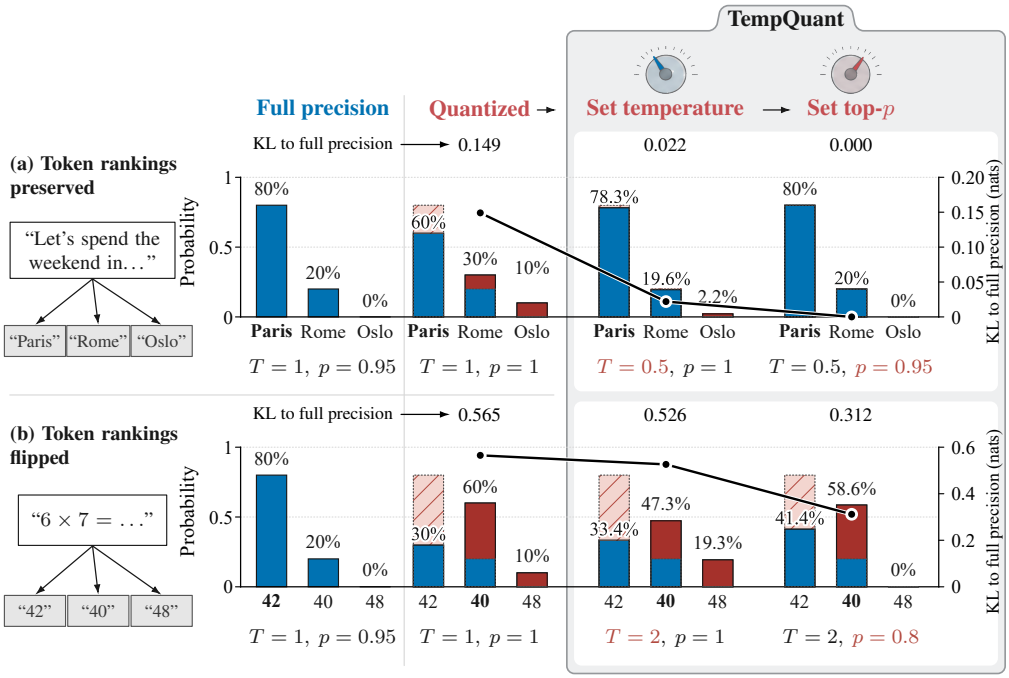

Figure 1: **Temperature and top-p correction.** Illustrative three-token distributions show how decoding adjustments reduce forward KL to full precision when quantization preserves or changes rankings.

output and layerwise supervision to fit a single temperature (Luo et al., 2026). We do not select temperature and top-p for confidence calibration, but to preserve the behavior of a quantized model.

Temperature and generation behavior. Temperature affects the diversity and coherence of generated text. Holtzman et al. (2020) show that low-temperature sampling can reduce diversity and increase repetitive loops in open-ended generation. In a Llama-2 story-generation study, Peeperkorn et al. (2024) find that higher temperature is only weakly associated with novelty and more strongly associated with incoherence. These findings motivate measuring diversity, repetition, and task quality separately.

Behavioral effects of quantization. Quantization can change reasoning behavior as well as accuracy. Lotfi et al. (2026) find that aggressive quantization can increase overthinking errors, where models reach a correct intermediate answer and then abandon it. They reduce these errors with a penalty on overthinking markers. In dialogue evaluations, Li et al. (2024) document sentence-level repetition under 3-bit weight or KV-cache quantization and repeated-token failures at 2 bits. The effects vary by configuration: Liu et al. (2025) find response lengths close to BF16 under several settings, while more aggressive quantization can lengthen outputs. We measure response length and repetition alongside accuracy, using BF16 as our behavioral reference.

Quantization impacts on token probabilities. Wu et al. (2026) find that low-bit quantization can shrink decision margins (the logit differences between the first tokens of two candidate continuations) and introduce directional biases that reverse model preferences. A margin describes a particular choice, while entropy describes uncertainty over the full vocabulary; one does not determine the other. We study the related question of how quantization changes entropy and whether those changes can be understood as temperature shifts.

3 Method

Token distributions. The goal of TempQuant is to bring the probability distribution emitted at each step by the quantized model as close as possible to what the full-precision model emits after

applying the user's requested temperature and top-p. We compare the two models after the same prefix, meaning the same sequence of input tokens. To apply the decoding settings, we divide the logits by temperature and convert them to probabilities. Top-p retains the most probable tokens until their cumulative probability reaches the cutoff, then rescales the retained probabilities to sum to one (Holtzman et al., 2020).

Let $u_t, z_t \in \mathbb{R}^V$ be the full-precision and quantized logits at prefix c_t, and let $\mathcal{N}_r$ denote top-p filtering with cutoff r. The full-precision model, with temperature $\tau > 0$ and cutoff ρ, emits the probability distribution

$$p_t^B = \mathcal{N}_\rho(\mathrm{softmax}\Big(\frac{u_t}{\tau}\Big)). \tag{1}$$

We fix the reference cutoff at $\rho = .95$ in our experiments. TempQuant predicts quantized-model settings T_t and ρ_t^Q, which give the corrected probability distribution

$$q_t = \mathcal{N}_{\rho_t^Q}(\mathrm{softmax}\left(\frac{z_t}{T_t}\right)). \tag{2}$$

We train TempQuant to predict settings that bring this quantized distribution closer to the full-precision reference at the same prefix c_t.

Predicting temperature and top-p. During inference, TempQuant needs to make these predictions using only information available from the quantized model. At each generation step, two MLP heads use 42 features of the current quantized logits that account for the requested temperature. This input combines 32 normalized, rank-ordered logits with ten summaries of logit scale, leading-token gaps, entropy, and probability concentration. The controller predicts a temperature correction factor and a top-p cutoff. Multiplying the requested temperature by the correction factor gives the temperature used for sampling. Appendix A gives the exact feature construction, output conversion, and training details.

We borrow the two-head design from AutoDeco (Wang et al., 2025), which also predicts temperature and top-p at each step and passes the predicted temperature to the top-p head. Changing temperature changes token probabilities and can change which tokens a given cutoff retains. With $x_t(\tau)$ denoting the features and κ_t the correction factor, we write the predictions as

$$\kappa_t = F_\theta(x_t(\tau)), \qquad \rho_t^Q = G_\phi(x_t(\tau), \log \kappa_t), \qquad T_t = \tau\kappa_t. \tag{3}$$

The top-p prediction depends on both the features and the temperature prediction. Figure 2 shows how the heads control the sampler.

Training objective. During training, we give both models the same prompt and text generated so far. TempQuant learns to choose temperature and top-p settings that make the quantized model's next-token probabilities closer to those of the full-precision model. Following the soft-target approach of LLM-QAT (Liu et al., 2023), we supervise with the full reference distribution rather than only the token sampled during generation.

One difficulty is that ordinary top-p filtering keeps or excludes certain tokens discretely. Small changes in the cutoff can leave the retained tokens unchanged, so this operation does not provide the gradients needed to learn the cutoff. Following AutoDeco, we use a smooth approximation during training that gradually downweights tokens around the cutoff. This allows the loss to update the top-p head along with the temperature head. The full-precision reference still uses ordinary top-p filtering.

To measure the difference between the quantized and full-precision probabilities, we compare forward KL, reverse KL, and symmetric JSD. Appendix E.1 illustrates these three measurements. Agarwal et al. (2024) show that the choice of divergence can matter when a student cannot fully represent a teacher's distribution. Our method can be thought of as a restricted form of knowledge distillation (Hinton et al., 2015), where the student is the quantized model and the teacher is the full-precision model. We can only change the student's temperature and top-p, so it may not be able to reproduce the teacher's distribution. We therefore train controllers with each of the three divergences, using the same architecture and training budget, and evaluate the downstream performance of each. We

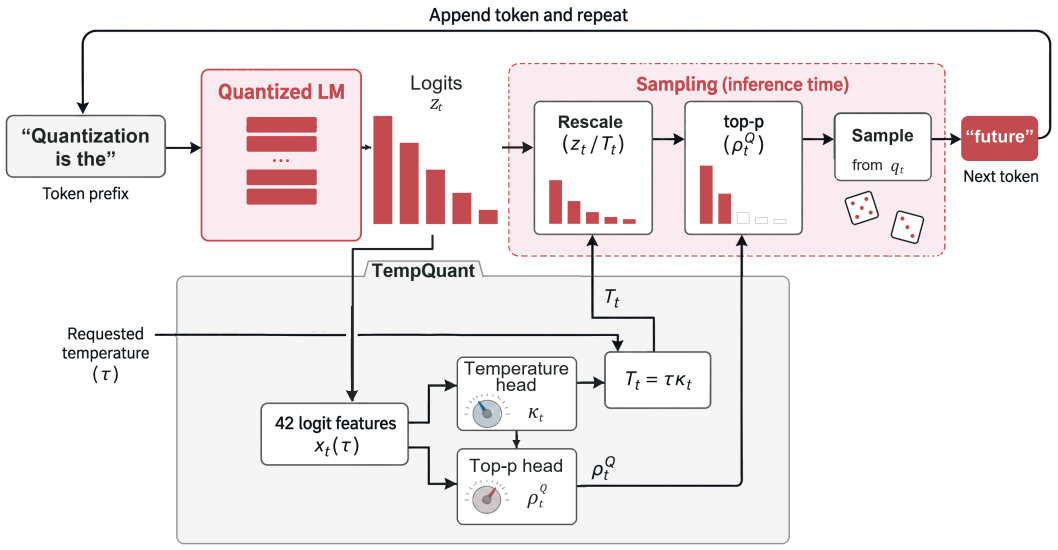

Figure 2: **TempQuant during inference.** Quantized-logit features determine temperature and top-p. Dice indicate sampling from the corrected probabilities, not uniform selection.

report all three objectives without selecting a single winner. Appendix D.1 gives the full fidelity comparison.

During training, we must set the temperature for the full-precision model. The requested temperature changes the reference probabilities that our controller aims to match, as well as the quantized-logit features it uses. To expose the controller to a range of these distributions, we train one shared controller at three requested temperatures: 0.3, 0.6, and 1.0. Let $\widetilde{q}_t$ denote the corrected quantized probabilities under the smooth approximation. The common training objective is

$$\min_{\theta,\phi} \; c_D \sum_t w_t D(p_t^B, \widetilde{q}_t) + \lambda \mathcal{R}(\theta, \phi), \tag{4}$$

where D is the chosen discrepancy and w_t weights the prefix–temperature pairs equally across the three requested temperatures. The regularizer $\mathcal{R}$ discourages large variation in the predicted settings across training prefixes. The loss scale c_D, regularization weight λ, and exact penalty are specified in Appendix A. Appendix B reports the history, on-policy, and training-temperature experiments.

Another issue is that we need to measure divergence after applying top-p. The quantized model can retain a different token set from the full-precision model. Ordinary forward KL becomes infinite when a token with positive BF16 probability has zero quantized probability. Reverse KL has the same problem with the roles exchanged. We address this in the KL recipes by mixing each distribution with a small uniform distribution before computing the loss. This gives every valid token positive probability. It defines a smoothed discrepancy, whose value can depend on the mixture weight. Note that this is only applied to forward and reverse KL, as JSD remains finite without this mixture. Appendix E.2 defines the objectives and the numerical treatment.

We also test whether training with smoothed KL reduces unsmoothed forward KL before applying top-p.

Both language models and their vocabulary heads remain frozen. We update only the two prediction heads, jointly, including gradients through the temperature prediction supplied to the top-p head.

Generation. At each step, we compute the 42 quantized-logit features for the requested temperature τ. TempQuant predicts the correction factor κ_t and top-p cutoff ρ_t^Q, giving sampling temperature $T_t = \tau\kappa_t$. We divide the raw logits by T_t once, apply native hard top-p filtering with ρ_t^Q, and sample the next token. This token is appended to the prefix for the next step. No full-precision model is needed during inference. Figure 2 shows this inference path.

4 EXPERIMENTS

We organize the evaluation around three questions: whether learned correction preserves behavior and downstream performance, whether quantization produces a temperature-like shift, and how much error TempQuant can correct.

The primary configuration is the post-trained Qwen3-4B (Yang et al., 2025) with native microscaling four-bit floating-point (MXFP4) weights and activations (W4A4), and BF16 key–value (KV) storage. BF16 and uncorrected quantization use matching requested temperatures .3, .6, and 1.0, with top-p = .95. We evaluate each trained controller at all three temperatures. Since we use stochastic sampling, we use four responses per item on GSM8K, four on MATH-500 except for two matched trials at 0.6, and eight on AIME-120, GPQA-Diamond, and XSum. Appendices C and D give evaluation details and full results.

4.1 DOES CORRECTION PRESERVE MODEL BEHAVIOR AND PERFORMANCE?

Distillation fidelity need not translate into better task performance (Stanton et al., 2021). We distinguish task accuracy or quality against dataset references from similarity to BF16 responses, and from diversity, repetition, and length relative to BF16. We report all three training objectives to show their tradeoffs, without presenting one as a universal winner.

Task performance. Table 1 reports the completed downstream results.

	Accuracy (%)				XSum quality ROUGE-2	Average
Policy	GSM8K	MATH	AIME	GPQA	vs. human	(5 tasks)
Temperature (τ) = 0.3						
BF16	94.47	95.40	65.6	39.8	6.76	60.41
Native MXFP4	93.67	91.95	43.2	**41.5**	**5.62**	55.19
TempQuant, forward KL	**93.75**	92.48	**47.9**	40.1	5.56	**55.96**
TempQuant, reverse KL	93.51	92.15	44.1	39.4	5.58	54.96
TempQuant, JSD	93.56	**92.70**	46.0	39.5	5.60	55.48
Temperature (τ) = 0.6						
BF16	94.47	95.30	68.2	41.5	6.65	61.24
Native MXFP4	93.27	92.50	45.8	39.3	5.59	55.29
TempQuant, forward KL	**93.67**	92.85	46.9	**42.0**	5.53	56.20
TempQuant, reverse KL	93.18	92.80	44.5	39.5	**5.69**	55.13
TempQuant, JSD	93.66	**93.25**	**48.0**	40.7	5.65	**56.26**
Temperature (τ) = 1.0						
BF16	95.02	95.55	69.3	42.7	6.24	61.76
Native MXFP4	93.65	92.55	48.1	39.6	5.65	55.91
TempQuant, forward KL	**93.91**	**93.20**	46.6	**40.9**	5.57	56.04
TempQuant, reverse KL	93.48	93.15	**48.8**	39.3	**5.68**	56.07
TempQuant, JSD	93.88	92.68	48.2	**40.9**	5.62	**56.26**

Table 1: **Downstream performance.** Learned rows average both fits. MATH uses missing-grade bound midpoints (Appendix D.5). Average equally weights all five unrounded scores, mixing accuracy and ROUGE. Bold marks the highest quantized estimate; green marks gains over native. Neither indicates significance; BF16 is a separate reference.

Effects vary across tasks and temperatures. Nominal paired intervals support gains for forward KL on GPQA at 0.6, for forward KL and JSD on AIME at 0.3, and for JSD on AIME at 0.6. Some AIME estimates worsen, and all learned policies remain well below BF16 there. Pooled across temperatures, JSD improves AIME; the other pooled AIME and all pooled GPQA intervals include zero (Table 35). All GSM8K and XSum quality intervals also include zero, which does not establish equivalence.

MATH estimates improve over native but remain below BF16. They assign half credit to three ungradable reference answers; full bounds are retained in Appendix D.5. The 0.6 comparison uses two complete matched trials because some other responses were lost. These descriptive results do not select a winning objective. Paired intervals elsewhere condition on the two fitted controllers and are not multiplicity-adjusted.

Matched decoding controls. Table 2 compares forward-KL TempQuant with the available matched controls. Neither comparison establishes an accuracy benefit over its control. The temperature-only contrast with native also has an interval containing zero (Appendix D.6).

Temperature (τ) = 0.3		Temperature (τ) = 0.6	
Policy	Accuracy (%)	Policy	Accuracy (%)
BF16	39.8	BF16	41.5
Native MXFP4	**41.5**	Native MXFP4	39.3
Fitted global temperature	39.5	Learned temperature only	41.6
Learned temperature/top-p	40.1	Learned temperature/top-p	**42.0**

Table 2: **Matched GPQA controls.** Forward-KL-trained policies; learned rows average both fits. Bold marks the highest quantized estimate. Counts and paired intervals: Appendix D.6.

Similarity to BF16 responses. Table 3 measures lexical similarity to BF16.

	Temperature (τ)		
Policy	0.3	0.6	1.0
BF16/BF16	**92.70**	**86.37**	**78.00**
Native MXFP4	45.69	48.72	50.57
TempQuant, forward KL	44.41	46.16	48.20
TempQuant, reverse KL	45.66	48.90	51.52
TempQuant, JSD	45.85	48.13	49.82

Table 3: **XSum reference similarity: BLEU vs. BF16.** BF16/BF16 measures within-reference variation, not a theoretical maximum. Learned rows average both fits. Paired intervals: Table 29.

On XSum, forward-KL training lowers lexical overlap with BF16 at all three temperatures, despite reducing forward KL on the saved prefixes. Reverse-KL training gives a small overlap increase at 1.0, while JSD lowers overlap there. The nominal paired intervals are in Table 29. Lexical overlap is not semantic agreement or summary quality; these measurements describe different aspects of reference preservation.

Diversity, repetition, and length. On XSum, forward-KL training produces more diverse responses but moves the measured diversity farther from BF16. Reverse KL gives closer agreement on that statistic (Table 30). Outputs are short, repeated four-grams are rare, and no quantized response reaches the cap. These results do not establish recovery from repetitive loops.

On GPQA at $\tau = 0.3$, forward-KL controllers avoid the capped responses seen under native quantization, but have lower accuracy. Fewer caps do not by themselves show better answers. Reverse KL increases the cap-rate estimate, while JSD decreases it (Table 32). Caps record budget exhaustion, not necessarily loops; they are rare at $\tau = 1.0$ (Appendix D.9).

4.2 Does quantization change the temperature of your model?

BF16 and native MXFP4 each generate their own responses at deployed temperatures .3, .6, and 1.0, with top-$p = .95$. Figure 3 uses 198 GPQA questions and eight trials per model and temperature: 9,504 responses in total. Entropy is replay-derived and provisional because agreement with the historical generation path is not established. MXFP4 has slightly higher mean entropy before and after top-p at all three temperatures. All nominal pre/post-filter paired intervals exclude zero, without multiplicity adjustment. This does not support general cooling for this configuration. The separate shared-prefix diagnostic measures conditional prediction differences (Appendix D.11).

The *generation-level entropy-equivalent BF16 temperature* matches mean entropy using each model's own trajectories at its respective setting. The sparse BF16 grid does not resolve a precise inverse, and MXFP4 entropy at 1.0 exceeds its range. Changing trajectories can also make mean entropy nonmonotonic. We therefore report no temperature matches. Matching entropy would not imply matching probabilities, rankings, answers, or exact logit scaling (Appendix D.12).

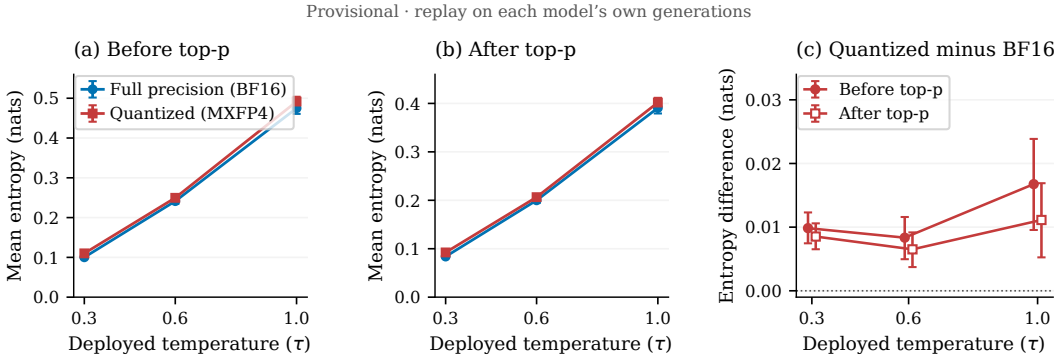

Figure 3: **Entropy during each model's own generation (provisional).** Mean next-token entropy and paired MXFP4-minus-BF16 differences, estimated by replay. Bars show nominal 95% question-bootstrap intervals. Question-balanced weighting and replay limitations are described in Appendix D.12.

4.3 How much quantization error can we correct through TempQuant?

Global temperature correction. Fitted globals use the same training data, objective, and bounds as the adaptive controller. Both global controls reduce development smoothed forward KL (Table 4). Each global is fitted for its requested temperature, while the learned controller is trained across all three. These fitted globals are not certified optima.

	Temperature (τ)		
Policy	0.3	0.6	1.0
Native MXFP4	1.3459	0.9117	0.7486
Fitted global temperature	1.0215	0.7124	0.4158
Fitted global temperature/top-p	0.9927	0.6885	0.3984
Learned temperature only	0.5117	0.2960	0.2378
Learned temperature/top-p	0.4413	0.2720	0.2106

Table 4: **Fitted global and adaptive correction.** Smoothed forward KL (nats) after top-p, using the same development prefixes and forward-KL training. Learned rows average both fits. Full settings and tradeoffs: Appendix D.2.

Per-token temperature oracle. We next ask how much mismatch temperature could remove if BF16 information were available at every prefix. On the saved development prefixes, the temperature oracle minimizes forward KL within the allowed range while leaving the quantized distribution unfiltered. It removes 65.91% of the finite baseline KL (Table 5).

Joint temperature and top-p oracle. Allowing the oracle to choose both settings removes 81.56% of the same baseline KL. The additional control lets the oracle exclude tokens as well as change probability concentration. These are mathematical-oracle results on development prefixes, not learned-controller scores or native-runtime guarantees.

Learned correction. Without BF16 access at deployment, forward-KL TempQuant removes 47.2–60.5% of the fitted joint global's remaining smoothed forward KL across the three settings. Learning top-p lowers that metric by 8.1–13.8% beyond the separately trained temperature-only control, using two-fit means. JSD, reverse KL, and total variation worsen relative to native quantization. The other objectives give smaller improvements over their matched globals (Appendix D.2). These development ratios do not select a downstream winner.

Student policy	Forward KL (nats)	Reduction (%)
Uncorrected baseline ($T = .6$, $p = 1$)	0.3201	0.0
Per-prefix temperature oracle ($p = 1$)	0.1091	65.9
Per-prefix temperature and top-p oracle	0.0590	81.6

Table 5: **Development forward-KL oracle recovery.** Reductions use the finite, unfiltered baseline against the same post-filter BF16 target. Uncertainty and tie sensitivity: Appendix E.6.

The same forward-KL recipe also improves development fidelity for Gemma 4 31B: smoothed forward KL falls from 0.3391 for native MXFP4 to 0.0883 averaged over the two learned fits, compared with 0.1944 for the fitted joint global. JSD, reverse KL, and total variation worsen. This separate fixed-0.6 study uses Gemma-specific training. Its 32-question, one-trial downstream pilot shows no learned accuracy gain over native on either MATH or GPQA (Appendix D.3).

Original divergence removed and additional global-to-oracle recovery have different denominators (Equation 9). The oracle and learned comparisons differ in target conventions, metrics, baselines, and arithmetic, so they cannot form one recovery sequence. Local oracle optimality also does not bound benchmark accuracy.

5 Limitations

Temperature and top-p cannot repair rankings. Smooth training differs from hard-filtered generation, and mathematical oracles do not certify native-runtime optima. Lexical overlap, length, and caps do not establish semantic preservation, looping, or abandoned correct answers. We do not measure difficulty effects or controlled deployment overhead. Downstream controls cover two GPQA panels; external methods and online-oracle generation are outside this evaluation.

6 Conclusion

TempQuant learns per-token temperature and top-p to match a requested BF16 policy without updating language-model weights. Adaptive correction improves its training metric beyond fitted globals, with cross-metric tradeoffs. Downstream gains vary by task and temperature, and better distributional fidelity does not uniformly improve response agreement or quality. The provisional own-generation study finds higher entropy under MXFP4, not general cooling. Decoding adjustments can recover distributional fidelity, but that recovery alone does not predict task performance. Our AI-use disclosure is in Appendix F.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649v3.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. *arXiv preprint arXiv:2407.09141*, 2024. URL https://arxiv.org/abs/2407.09141v1.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323v2.

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. URL https://arxiv.org/abs/2103.03874.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1904.09751v2.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951. ISSN 0003-4851. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. arXiv preprint arXiv:2309.06180, 2023. URL https://arxiv.org/abs/2309.06180.

Sihwa Lee, Janghwan Lee, Donghoon Yoo, Jae Gon Kim, Hanyul Ryu, Soojung Ryu, and Jungwook Choi. ReSET: Accurate latency-critical NVFP4 reasoning via step-aware temperature scaling. *arXiv preprint arXiv:2606.13233*, 2026. URL https://arxiv.org/abs/2606.13233v1.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models, 2024. URL https://arxiv.org/abs/2402.18158v1.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models, 2025. URL https://arxiv.org/abs/2504.04823v2.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models, 2023. URL https://arxiv.org/abs/2305.17888.

Sanae Lotfi, Polina Kirichenko, Steven Li, and Zechun Liu. Quantized reasoning models think they need to think longer, but they do not, 2026. URL https://arxiv.org/abs/2606.00206v1.

Beier Luo, Shuoyuan Wang, Sharon Li, and Hongxin Wei. Your pre-trained LLM is secretly an unsupervised confidence calibrator. *arXiv preprint arXiv:2505.16690*, 2025. URL https://arxiv.org/abs/2505.16690v5.

Beier Luo, Cheng Wang, Hongxin Wei, Sharon Li, and Xuefeng Du. Unlocking the pre-trained model as a dual-alignment calibrator for post-trained LLMs. *arXiv preprint arXiv:2601.04277*, 2026. URL https://arxiv.org/abs/2601.04277v1.

Pierre-Alexandre Mattei and Bruno Loureiro. The well-tempered classifier: Some elementary properties of temperature scaling. *arXiv preprint arXiv:2602.14862*, 2026. URL https://arxiv.org/abs/2602.14862v2.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is temperature the creativity parameter of large language models?, 2024. URL https://arxiv.org/abs/2405.00492v1.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Zi-Hao Qiu, Siqi Guo, Mao Xu, Tuo Zhao, Lijun Zhang, and Tianbao Yang. To cool or not to cool? temperature network meets large foundation models via DRO. *arXiv preprint arXiv:2404.04575*, 2024. URL https://arxiv.org/abs/2404.04575v3.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. arXiv preprint arXiv:2311.12022, 2023. URL https://arxiv.org/abs/2311.12022.

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory Wornell, and Soumya Ghosh. Thermometer: Towards universal calibration for large language models. *arXiv preprint arXiv:2403.08819*, 2024. URL https://arxiv.org/abs/2403.08819v2.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. A thorough examination of decoding methods in the era of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8601–8629. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.489. URL https://arxiv.org/abs/2402.06925v1. Numerical comparisons refer to arXiv:2402.06925v1, Section 5.3 and Tables 3–4.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6906–6919. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/376c6b9ff3bedbbea56751a84fffc10c-Paper.pdf.

Zhichao Wang, Dongyang Ma, Xinting Huang, Deng Cai, Tian Lan, Jiahao Xu, Haitao Mi, Xiaoying Tang, and Yan Wang. The end of manual decoding: Towards truly end-to-end language models, 2025. URL https://arxiv.org/abs/2510.26697v1.

Zekun Wu, Swati Dhiman, and Adriano Koshiyama. Which decisions low-bit quantization breaks, and how to predict them. arXiv preprint arXiv:2608.06564, 2026. URL https://arxiv.org/abs/2608.06564v3.

Johnathan Xie, Annie S. Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating language models with adaptive temperature scaling. *arXiv preprint arXiv:2409.19817*, 2024. URL https://arxiv.org/abs/2409.19817v1.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

A Controller architecture and training

This section specifies the controller in Figure 2.

A.1 Architecture and inputs

Both the temperature head and the top-p head use the current quantized logits. They share the same 42-feature input, but have separate weights. The top-p head also receives the logarithm of the temperature head's internal output h, allowing gradients to flow back to the temperature head.

Model and quantization. The post-trained model and tokenizer are `Qwen/Qwen3-4B` (Yang et al., 2025). Quantization preserves the tokenizer and chat template. Linear projection weights use static MXFP4 and their inputs use dynamic MXFP4, in groups of 32 with shared E8M0 scales. Embeddings, normalization parameters, the tied vocabulary head, and KV storage remain BF16. Model serving uses vLLM (Kwon et al., 2023).

Controller architecture	
Model	Qwen3-4B, native MXFP4 W4A4, BF16 KV
Reference	BF16 temperatures $0.3, 0.6, 1.0$, top-$p = .95$
Input	Ten distribution summaries followed by 32 ranked logit features
Temperature head	$42 \to 1398 \to 16 \to 1$, Tanh hidden activations
Top-p head	$43 \to 1375 \to 16 \to 1$, Tanh
Trainable parameters	165,048
Temperature output	$h \in (.05, 50)$
Top-p output	$\rho^Q \in (0, 1)$
Sampling temperature	$T = \tau\kappa$, the requested temperature τ multiplied by the correction factor

Table 6: Architecture of the shared-temperature controller. For each training objective, we train two controllers with different random seeds and average their evaluation results.

During generation, TempQuant uses only the quantized model and the trained heads. Computing the input features requires processing the full vocabulary at every step, so the small controller size alone does not establish low overhead. Runtime and memory overhead are not measured in this study.

A.2 Input features

Table 7 lists the 42 logit features we feed into the controller. We adjust the logits for the requested temperature before computing the features. Probability features describe the distribution before correction or top-p filtering. Each feature is normalized using its training-set mean and standard deviation, which remain fixed during evaluation.

In the table, $\widetilde{z}$ denotes logits scaled by τ_0/τ and then shifted so their maximum is zero, with $\tau_0 = 0.6$. Its kth largest value is r_k. The scale s is the standard deviation of all logits in $\widetilde{z}$, with a small lower bound to avoid division by zero. We use $v = \mathrm{softmax}(\widetilde{z}/0.6)$, with $v_{(k)}$ denoting its kth largest probability. Centering makes coordinates 6 and 11 zero; both are retained.

Coordinate	Feature	Definition
1	Gap between the first- and second-ranked logits	$r_1 - r_2$
2	Gap between the first- and eighth-ranked logits	$r_1 - r_8$
3	Gap between the first- and 32nd-ranked logits	$r_1 - r_{32}$
4	Logit sample standard deviation	$\mathrm{std}(\widetilde{z})$
5	Mean logit	$V^{-1}\sum_i \widetilde{z}_i$
6	Maximum logit after centering	$r_1 = 0$
7	Entropy (nats)	$H(v) = -\sum_i v_i \log v_i$
8	Largest token probability	$v_{(1)}$
9	Probability mass of the eight most likely tokens	$\sum_{k=1}^{8} v_{(k)}$
10	Probability mass of the 32 most likely tokens	$\sum_{k=1}^{32} v_{(k)}$
11–42	The 32 largest logits divided by their scale	$r_k/s, \quad k = 1, \ldots, 32$

Table 7: The controller's 42 input features before fitted normalization. The first ten summarize the distribution; the last 32 describe its leading logits. Probability features use the requested temperature, before correction or top-p filtering.

A.3 Output ranges and starting values

The heads convert their predictions into a positive temperature and a top-p cutoff between zero and one. A sigmoid function σ keeps the outputs within their allowed ranges. For head outputs $a_\theta(x)$ and $b_\phi(x, \log h)$,

$$\log h = \log h_{\min} + \sigma(a_\theta(x) + c_T) \log\left(\frac{h_{\max}}{h_{\min}}\right),$$
$$\rho^Q = \sigma(b_\phi(x, \log h) + c_p), \qquad (h_{\min}, h_{\max}) = (0.05, 50). \tag{5}$$

The temperature head predicts h for the rescaled logits. Dividing by 0.6 gives the correction factor $\kappa = h/0.6$, so the sampling temperature is $T = \tau\kappa$. Thus $h = 0.6$ leaves the requested temperature unchanged; larger values raise it and smaller values lower it. The top-p output ρ^Q sets the cumulative probability cutoff for retaining tokens.

To connect these outputs to Equation 3, let $f_\theta(x) = h$ and $g_\phi(x, \ell) = \sigma(b_\phi(x, \ell) + c_p)$. Then

$$F_\theta(x) = \frac{f_\theta(x)}{\tau_0}, \qquad G_\phi(x, \ell) = g_\phi(x, \ell + \log \tau_0). \tag{6}$$

The shift by $\log \tau_0$ means the cutoff head receives $\log h$. During training, gradients through this input also update the temperature head.

Before training, the heads return the same correction factor and cutoff for every input. These starting values come from an earlier global temperature/top-p pair fitted on training data with forward KL. The fixed constants c_T and c_p set these values. Hidden layers are randomly initialized, and the final layers start at zero. Within each seed, all objectives start from identical weights. Exact initialization settings and numerical details are provided in the accompanying code.

A.4 Training settings

Only the two heads are updated. Both language models, vocabulary heads, and input normalizer remain frozen. Table 8 gives the common optimization settings.

We evaluate the final controller from each run after 120 training updates, without selecting an earlier checkpoint using development results.

Action-variance regularization. We add a small penalty when predicted temperatures and top-p cutoffs vary widely across training examples at the same requested temperature. This encourages more consistent settings while allowing adaptation when it improves the match to BF16. The penalty

Training configuration	
Objectives	Smoothed forward KL, smoothed reverse KL, symmetric JSD
Loss scale c_D	$1/4$ for either KL objective; 1 for JSD
Uniform mixture η	10^{-4} for KL; none for JSD
Smooth top-p width	$\epsilon = .01$
Action-variance weight	$\lambda = .01$
Optimizer	Adam: $\beta_1 = .9$, $\beta_2 = .999$, $\epsilon_{\text{Adam}} = 10^{-8}$, weight decay 0; without AMSGrad
Learning rate	.001 for updates 1–30; .0001 for 31–120; optimizer state retained at the transition
Update / clipping	Full weighted pool of 6,912 rows, accumulated in chunks of 32; global gradient norm clipped to 1 before each update

Table 8: Training configuration shared by all six fits except for the stated objective and loss scale.

measures variation across examples, not changes between consecutive generation steps. With normalized prefix weights $\sum_t w_t = 1$ at each requested temperature,

$$\mathcal{R} = \frac{1}{3} \sum_{\tau \in \{0.3, 0.6, 1.0\}} \left[\sum_t w_t (\ell_{t,\tau} - \bar{\ell}_\tau)^2 + \sum_t w_t (v_{t,\tau} - \bar{v}_\tau)^2 \right], \tag{7}$$

where $\ell_{t,\tau} = \log h_t(\tau)$, $v_{t,\tau} = \operatorname{logit} \rho_t^Q(\tau)$, and bars are weighted means at that requested temperature. Using $\log \kappa$ instead of $\log h$ gives the same variance, since they differ by a constant.

Smooth top-p. During training, we use a smooth top-p approximation so gradients can update the cutoff. Sort logits in ascending order and let b_i be their probabilities after temperature scaling, with $C_i = \sum_{j \le i} b_j$. For $\epsilon = 0.01$,

$$\log g_i = \log \sigma\left(\frac{C_i - (1 - \rho^Q)}{\epsilon}\right) - \log \sigma\left(\frac{C_i}{\epsilon}\right), \qquad \widetilde{q}_i = \frac{b_i g_i}{\sum_j b_j g_j}, \tag{8}$$

with $g_i = 1$ for the highest-ranked token. BF16 targets, evaluation, and generation use ordinary top-p.

A.5 Training data

We train on 24 questions from GSM8K (Cobbe et al., 2021) and 24 from MATH (Hendrycks et al., 2021). A separate development set contains 32 questions from each dataset. All questions come from the datasets' training splits, excluding MATH-500 test questions. We screen for overlapping question text and select questions without looking at model outputs.

For each question, we collect six responses: two from BF16, two from uncorrected quantization, and two from quantization using a previously trained TempQuant controller. BF16 and uncorrected quantization use temperature 0.6 and top-p 0.95. The earlier controller's weights remain unchanged during data collection.

We randomly select eight distinct points from each training response and two from each development response. At each point, both models receive the same question and generated text so far, allowing us to compare their next-token distributions. This gives 2,304 training prefixes and 768 development prefixes. We average over the selected points within each response, giving equal weight to the responses, generation methods, questions, and tasks.

We reuse these saved prefixes when training at requested temperatures 0.3, 0.6, and 1.0, giving each temperature equal weight. We do not generate new responses as training progresses. Because development results helped guide our method choices, they are not an independent test of the final method.

A.6 COMPARING CONTROLLER INPUTS

We compared logit features with hidden-state inputs and with additional projections of the full logits that retain token identity. This earlier study used JSD training at a fixed BF16 temperature of 0.6 and top-p 0.95. The alternatives differ in size and preprocessing, so they do not isolate the effect of input representation perfectly. Their configurations are provided in the accompanying code.

Policy / input	Pooled JSD	GSM8K	MATH
Native MXFP4	0.0430	0.0495	0.0365
Global temperature	0.0417	0.0481	0.0352
Global temperature/top-p	0.0417	0.0481	0.0354
Full hidden state	0.0429	0.0473	0.0385
42 logit features, small heads	0.0411	0.0469	0.0353
42 logit features, larger heads	0.0407	0.0462	0.0352
42 features + fixed full-logit projection	0.0409	0.0463	0.0355
42 features + learned full-logit projection	0.0409	0.0462	0.0355
Paired comparison		JSD change (10^{-3} nats) [95% CI]	
Larger − small logit heads		$-0.38\ [-0.86, +0.03]$	
Learned projection − larger logit heads		$+0.17\ [-0.41, +0.80]$	

Table 9: **Input representation comparison.** JSD (nats) after ordinary top-p on 768 reused development prefixes. Learned rows average two fits after 120 updates; global controls were fitted with JSD on training data. Green marks reductions from native, not significance. Paired intervals follow Appendix C.3 and are not selection-adjusted. These fixed-reference results are separate from the shared-temperature study.

Logit features gave lower mean JSD than hidden-state inputs. Neither the larger heads nor the learned projection showed a clear incremental benefit in the paired comparison. These results support using logit features, without establishing an optimal head size.

B CONTROLLER DESIGN AND TRAINING EXPERIMENTS

We tried adding history, refreshing training responses, and changing the training temperature. All three studies used an earlier temperature-only TempQuant controller, with a BF16 target temperature of 0.6 and fixed top-p 0.95. Only a small additional correction head was trained; the base temperature head and both language models stayed fixed. The findings are limited to this earlier design.

The comparisons reuse development questions. Paired 95% intervals resample whole questions and are not adjusted for selection. Numerical consistency across the generation and replay paths was not fully validated.

B.1 ADDING HISTORY

Inspired by ReSET's use of recent reasoning context (Lee et al., 2026), we added summaries of past entropy and top-two logit margins to the current features (Figure 4). A small residual head used these summaries to adjust the base temperature. The matched control had the same architecture but its history inputs were masked. History used past quantized-model entropy and margins, with no BF16 values as inputs.

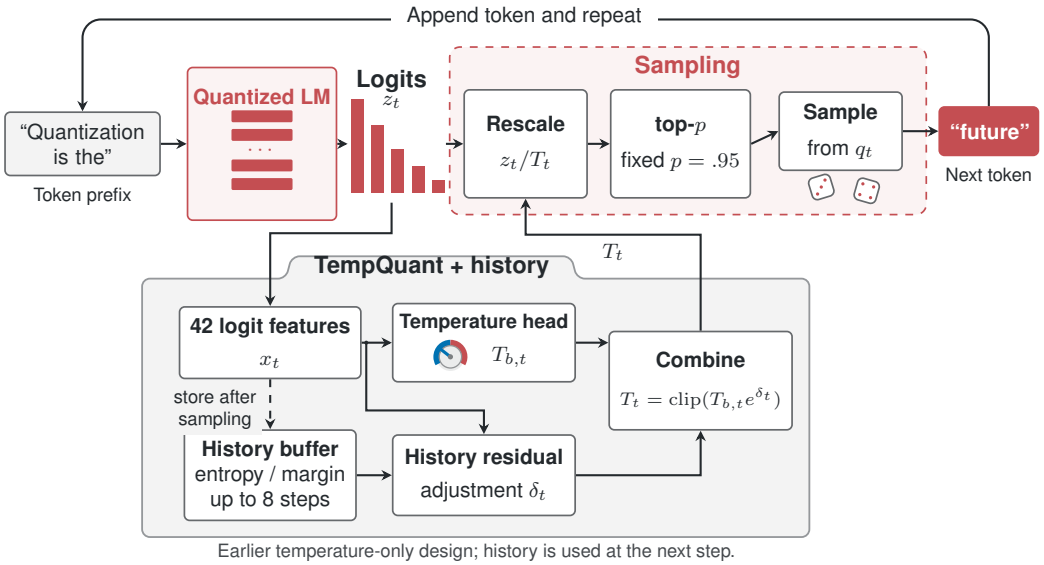

Figure 4: **Earlier TempQuant design with a history residual.** Seven summaries of up to eight preceding positions let the residual adjust the base temperature. The dashed connection updates history after sampling; top-p stays fixed. Only the residual is trained. The prompt, token, and logits are illustrative.

History gave small gains on short traces, but worsened KL on long reasoning (Table 10), along with total variation. Prose KL gains were inconclusive; total variation improved over the current-only control but remained worse than native quantization. A small falling-entropy subgroup improved, although the matched transition comparison was inconclusive. Resetting history inputs also improved some results, weakening the evidence that past observations caused the gains. Overall, history did not help reliably.

B.2 UPDATING TRAINING RESPONSES

Following GKD's use of student-generated supervision (Agarwal et al., 2024), we tested whether training improves when new responses come from the latest controller (Figure 5). The control instead generated fresh responses using the initial controller. Both continued training on new responses with the same update budget.

Comparison / fit	Current-only KL	History KL	History − current (10^{-3} nats) [95% CI]
Short traces: summaries	0.2198	0.2189	-0.92 [—]
Short traces: full window	0.2198	0.2227	$+2.88$ [—]
Long reasoning / fit 1	0.1358	0.1385	$+2.71$ $[+1.05, +3.99]$
Long reasoning / fit 2	0.1359	0.1404	$+4.46$ $[+1.93, +6.34]$
Prose / fit 1	0.4460	0.4464	$+0.40$ $[-1.23, +2.95]$
Prose / fit 2	0.4459	0.4458	-0.05 $[-1.32, +1.71]$

Table 10: **Adding history.** Forward KL before top-p; positive differences favor the current-only control. Short traces average two selected fits on 256 prefixes from eight questions; a pooled interval is unavailable. Long reasoning uses 709 positions from 16 questions; prose uses 512 from 32. Each cohort weights questions equally. Post-filter mean KL was infinite for every policy in both transfer cohorts because of missing support. All prose responses reached the token cap.

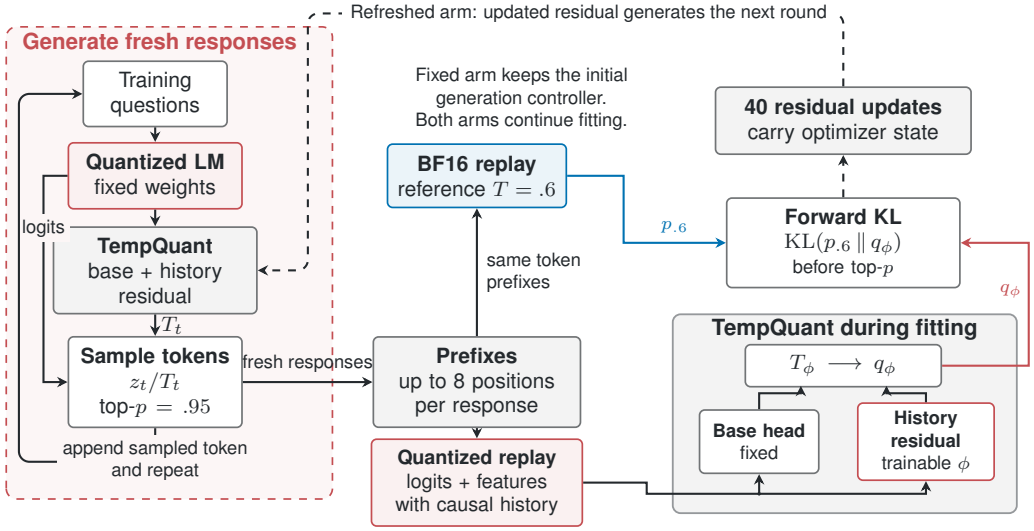

Figure 5: **Blockwise on-policy continuation of the earlier controller.** The quantized model generates freely. Both models are then evaluated on selected prefixes to train the residual. This repeats for three rounds of 40 updates. The refreshed condition uses the updated controller for the next round; the fixed condition keeps the initial generating controller. Both language models and the base temperature head remain fixed.

Training responses	Fixed	Refreshed	Refreshed − fixed [95% CI]
Initial controller	0.2160 (shared reference)		—
Response set 1	0.2189	0.2178	-0.0011 $[-0.0064, +0.0039]$
Response set 2	0.2163	0.2274	$+0.0111$ $[+0.0050, +0.0167]$

Table 11: **On-policy continuation after 120 updates.** Forward KL (nats) before top-p on 256 prefixes from eight reused development questions. The two response sets use different generation seeds, not different optimizer initializations. Of the 32 development responses, 31 reached the token cap. Positive differences favor fixed-generation training.

Refreshing responses did not reliably improve matching. It slightly lowered KL in one response set and increased it in the other; only the increase had an interval excluding zero. Total variation was worse than the fixed-generation control in both sets. Neither approach improved the initial controller's mean KL, although both remained below the matched global-temperature controls.

B.3 Training at different temperatures

In their MNIST experiments, Hinton et al. (2015) found that a much smaller student worked better with lower distillation temperatures than with the high temperatures used for larger students. Our earlier controller also has limited flexibility: with the language-model weights fixed, it can only sharpen or flatten the quantized distribution. This motivated testing whether colder training would help.

During training, we multiplied both the BF16 target temperature and the corrected quantized temperature by the same factor, and scaled the KL loss by that factor squared. Evaluation removed this extra scaling and kept the BF16 target at 0.6. Thus the experiment changes training temperature, not the requested deployment temperature.

		Forward KL (nats)		Total variation	
Factor	BF16 training temperature	Set 1	Set 2	Set 1	Set 2
Initial controller		0.216	0.216	0.149	0.149
Native quantization		0.416	0.416	0.109	0.109
0.5	0.30	0.306	0.291	0.223	0.216
0.75	0.45	0.247	0.248	0.173	0.175
1 (standard)	0.60	0.227	0.227	0.142	0.140
1.5	0.90	0.232	0.231	0.128	0.127
2	1.20	0.236	0.237	0.123	0.123
KL change [95% CI]		Response set 1		Response set 2	
Coldest − standard		$+0.080$ $[+0.038, +0.117]$		$+0.064$ $[+0.024, +0.099]$	

Table 12: **Training at different temperatures.** Measurements before top-p after 120 updates, evaluated against the same BF16 temperature-0.6 target. Each training response set supplies 512 prefixes from 32 questions; each factor is fitted from the same initial controller. Evaluation uses the same development prefixes as Table 11. Response sets vary generation seeds, not optimizer initialization.

Colder training did not help. Both colder settings increased KL relative to standard training, with paired intervals excluding zero in both response sets. Warmer training reduced total variation but increased KL; its total variation remained worse than native quantization. No setting improved the initial controller's mean KL. The standard-training increase over the initial controller was inconclusive.

The coldest setting helped the small subset where the models disagreed on the top token, but hurt the larger agreement subset. This breakdown was descriptive, without subgroup intervals. Removing the loss multiplier in the warmest setting did not reverse the overall KL result.

C EVALUATION DETAILS

Prompts, dataset identifiers, scorer versions, and implementation settings are provided in the accompanying code. We describe the study-specific choices below.

C.1 BENCHMARKS AND COMPARISON METHODS

Table 13 gives the primary populations. Each requested temperature $(0.3, 0.6, 1.0)$ compares BF16, native quantization, and two separately trained controllers for each objective.

Task	Items	Trials	Output limit (tokens)
GSM8K test	1,319	4	32,768
MATH-500 test	500	2 or 4	32,768
AIME-120	120	8	32,768
GPQA-Diamond	198	8	32,768
XSum validation subset	128	8	512

Table 13: **Benchmark coverage and output limits.** Response trials per item, deployed policy, and temperature. MATH uses two complete matched trials at 0.6 and four otherwise; three reference answers are ungradable (Appendix D.5). Completion status is shown in the result tables. XSum uses a validation subset, not the full test split.

GSM8K, MATH-500, and GPQA-Diamond (Rein et al., 2023) use their full test populations. AIME-120 combines AIME90 with the 30 problems from 2025. All XSum (Narayan et al., 2018) tables use the same validation subset, selected by deterministic hash order after deduplication and screening for text and 13-word-gram overlap with calibration inputs. Selection uses neither reference summaries nor model outputs; empty articles remain eligible. Earlier evaluation exposure remains a limitation.

Fitted globals minimize the matched training divergence using a single temperature or temperature/top-p pair within the learner's bounds. The separately trained temperature-only ablation fixes the cutoff at 0.95. Its development comparison and the fitted globals are complete (Appendix D.2). The completed downstream panels compare global temperature on GPQA at 0.3 and adaptive temperature only at 0.6 (Appendix D.6). Broader downstream control comparisons and accuracy-tuned controls are outside the reported evaluation.

We compare all three training objectives without selecting one from the downstream scores. The temperature-only ablation and Gemma study use forward KL as the documented fallback, not as an empirical winner.

C.2 GENERATION AND SCORING

Generation seeds match by task, question, and trial across policies and temperatures. Each model still generates its own response. Accuracy is average sampled correctness, not success on the best of several trials. Thinking mode is enabled for reasoning tasks and disabled for XSum. There is no repetition penalty or loop-triggered stop. GPQA additionally stops at `</s>`, excluding that string from scoring, and retains both questions with repeated distractors. XSum articles are limited to their first 4,096 tokens.

GSM8K and AIME use the Qwen2.5-Math scorer (Yang et al., 2024) on the whole saved output, including unclosed thinking blocks. MATH-500 uses Math-Verify on the last complete boxed answer, with a last-number fallback; the Qwen scorer is a sensitivity. Scoring MATH's whole output was a retrospective choice. GPQA uses LM-eval's flexible extraction, with strict extraction as a sensitivity.

XSum quality is ROUGE-2 F1 against human summaries (Lin, 2004). Extraction removes surrounding whitespace, terminal chat markers, and one leading complete thinking block. Unclosed, nonleading, or multiple thinking blocks and embedded chat controls produce an invalid empty candidate. Extraction never uses the reference to select or trim a summary.

C.3 Behavior measures and uncertainty

Length includes all generated tokens, including stop-causing tokens; Len(K) reports thousands. Cap rate measures responses reaching the output limit. Policy-specific capped subsets are not matched causal comparisons.

BLEU against BF16 (Papineni et al., 2002; Post, 2018) uses opposite halves of each item's trials as references, averaging corpus BLEU over both directions and learned fits. It scores the entire emitted text, unlike XSum quality extraction. BF16/BF16 measures within-reference variation, not a maximum. Self-BLEU measures similarity among responses to the same item; its absolute deviation from BF16 is calculated per item and fit before averaging. Repetition is the fraction of within-response four-grams beyond their first occurrence. Outputs too short to contain a four-gram contribute zero and are counted separately. These lexical measures do not establish semantic agreement or looping.

Response-level scores average trials within items, then the two learned fits, then items equally. Paired bootstrap intervals resample whole questions or articles, retaining their policies, trials, fits, and temperatures together. Pooled development comparisons are stratified by task; corpus BLEU is recomputed in each resample. Equal-temperature summaries weight the three settings equally. Intervals condition on the two trained controllers and do not capture full training variability or correct development reuse. The reported downstream intervals are nominal, without study-wide multiplicity adjustment. The shared-prefix entropy study's separate four-contrast adjustment appears in Appendix D.11.

Capped responses and emitted failures remain scored. Empty or invalid XSum summaries receive zero ROUGE; missing responses and unresolved scores remain unknown. Quality with unknown outcomes uses fixed-denominator bounds and completed-output sensitivity. Behavioral summaries disclose their observed denominators without imputing missing text or length. Cap and length data can remain available when the stop subtype is unknown.

C.4 Measuring error reduction

Let R, G, L, O be mean divergence for native quantization, a training-fitted global control, the learner, and a per-prefix oracle. For matched targets, metrics, filtering, action bounds, arithmetic, and weights,

$$\text{original divergence removed} = 1 - \frac{L}{R}, \qquad \text{global-to-oracle recovery} = \frac{G - L}{G - O}. \tag{9}$$

The ratios require finite losses, $R > 0$, and $G > O$ beyond numerical tolerance. Incompatible historical measurements cannot fill these comparisons. A best-found oracle is not a certified bound.

Original divergence removed compares the corrected loss with the uncorrected baseline; global-to-oracle recovery measures how much of the global-to-oracle gap the learner closes. The reported forward-KL oracle uses a post-filter BF16 target, an unfiltered student baseline, and FP64 student arithmetic. Learned-controller evaluation uses a pre-filter BF16 target for pre-filter KL and native hard-filtered distributions for post-filter metrics, with uniform mixing only for smoothed KL. Its native baseline uses the requested cutoff. These different conventions prevent combining the results into a single recovery sequence.

D Additional experimental results

These results use the six controllers trained equally across requested temperatures .3, .6, and 1.0. They are separate from the fixed-reference architecture-selection study in Appendix A.6.

D.1 Comparing training losses

Using the same predictor architecture, we train on forward KL, reverse KL, and symmetric JSD. Table 14 compares the resulting probability distributions on the same development prefixes.

Forward-KL training removes 67.2–71.9% of smoothed post-filter forward KL across the three settings, while increasing total variation (TV). JSD training reduces hard-filtered JSD at all three settings. Reverse-KL training reduces its corresponding smoothed metric at .6 and 1.0; the interval at

Policy	Before top-p filtering	After top-p filtering			
	Forward KL (nats)	Smoothed forward KL (nats)	Smoothed reverse KL (nats)	JSD (nats)	Total variation
Temperature (τ) = 0.3					
Native	0.6343	1.3459	1.3924	0.0557	0.1044
Forward KL	0.2472	0.4413	1.9522	0.0560	0.1266
Reverse KL	0.6560	1.6281	1.3717	0.0585	0.1052
JSD	0.3388	0.7889	1.5043	0.0507	0.1053
Temperature (τ) = 0.6					
Native	0.3163	0.9117	0.8503	0.0430	0.0997
Forward KL	0.1827	0.2720	1.5915	0.0490	0.1287
Reverse KL	0.4689	1.3265	0.8003	0.0488	0.1048
JSD	0.2182	0.5579	0.9765	0.0409	0.1007
Temperature (τ) = 1.0					
Native	0.1903	0.7486	0.6576	0.0359	0.0964
Forward KL	0.1370	0.2106	1.0846	0.0373	0.1089
Reverse KL	0.3059	1.1971	0.5616	0.0422	0.1054
JSD	0.1666	0.4942	0.7322	0.0344	0.0965

Table 14: **Shared-temperature development fidelity.** Absolute divergence values on 768 positions from 64 reused development questions; learned rows average the two fits, not an ensemble. Post-filter metrics use ordinary top-p. Only columns labeled *smoothed* add the 10^{-4} uniform mixture to both distributions, separately from the differentiable top-p used in training. Reductions and paired intervals appear once, in Table 15. Matched global and temperature-only controls are in Appendix D.2.

.3 includes zero. These percentages use different metrics and cannot rank the recipes. Tables 15 and 16 give paired intervals and support measurements.

The support measurements show the tradeoff directly. Forward-KL training retains more of BF16's probability mass but also assigns more probability outside BF16's retained token set. Reverse-KL training moves these two quantities in the opposite directions. A better score under one divergence therefore does not mean a better match in every respect.

Reductions use same-temperature native quantization as the denominator, with paired intervals from Appendix C.3.

Table 16 shows the support tradeoffs at $\tau = .6$ and the average settings predicted by each controller.

D.2 Fitted globals and temperature-only correction

Tables 17–19 compare all fitted controls on the same 768 development prefixes. Global controls are fitted separately at each requested temperature for 240 updates. Learned controllers use both shared-temperature fits after 120 updates. The 0.6 globals reuse matching earlier captures; their warm starts were not compared with multiple initializations. These fitted controls are not certified optima.

Forward-KL training gives the largest reductions in smoothed forward KL, but increases reverse KL and total variation. JSD and reverse-KL training give smaller improvements in their own objectives over matched globals. The temperature-only ablation uses forward KL, the documented fallback objective; it does not establish which training loss is best. All percentages in the main text are descriptive ratios of aggregate losses. No new paired control-contrast intervals were computed. The original per-policy intervals do not test a paired difference.

Metric	Temperature (τ) 0.3	0.6	1.0
Forward-KL training			
Before top-p: forward KL	$+61.0$ $[+55.7, +64.4]$	$+42.2$ $[+32.1, +49.2]$	$+28.0$ $[+20.1, +34.0]$
Smoothed forward KL	$+67.2$ $[+55.9, +76.3]$	$+70.2$ $[+61.2, +77.1]$	$+71.9$ $[+65.1, +77.1]$
Smoothed reverse KL	-40.2 $[-65.5, -23.6]$	-87.2 $[-126.1, -61.6]$	-64.9 $[-92.4, -46.5]$
JSD	-0.5 $[-13.4, +8.7]$	-14.0 $[-27.4, -3.7]$	-3.7 $[-12.7, +3.6]$
Total variation	-21.2 $[-35.5, -10.7]$	-29.0 $[-41.3, -19.3]$	-13.0 $[-20.0, -7.1]$
Reverse-KL training			
Before top-p: forward KL	-3.4 $[-11.9, +3.4]$	-48.2 $[-50.8, -45.9]$	-60.7 $[-65.3, -57.0]$
Smoothed forward KL	-21.0 $[-39.3, -8.9]$	-45.5 $[-63.7, -30.9]$	-59.9 $[-86.6, -41.6]$
Smoothed reverse KL	$+1.5$ $[-0.3, +3.9]$	$+5.9$ $[+1.7, +11.4]$	$+14.6$ $[+7.9, +23.3]$
JSD	-5.0 $[-8.2, -2.2]$	-13.5 $[-17.5, -9.6]$	-17.5 $[-22.4, -13.2]$
Total variation	-0.7 $[-2.3, +1.1]$	-5.0 $[-7.5, -2.2]$	-9.3 $[-11.9, -6.7]$
JSD training			
Before top-p: forward KL	$+46.6$ $[+44.5, +48.0]$	$+31.0$ $[+27.2, +33.8]$	$+12.5$ $[+10.9, +13.7]$
Smoothed forward KL	$+41.4$ $[+31.4, +51.1]$	$+38.8$ $[+27.6, +50.0]$	$+34.0$ $[+22.1, +45.8]$
Smoothed reverse KL	-8.0 $[-16.2, -2.6]$	-14.8 $[-23.4, -9.0]$	-11.4 $[-17.1, -7.2]$
JSD	$+9.0$ $[+4.1, +12.8]$	$+4.9$ $[+1.5, +7.7]$	$+4.2$ $[+1.7, +6.6]$
Total variation	-0.8 $[-5.8, +2.9]$	-0.9 $[-3.7, +1.5]$	-0.1 $[-1.7, +1.3]$

Table 15: **Paired development reductions (%).** Each cell gives reduction from same-temperature native quantization [paired 95% interval]. All metrics except the first row of each block are measured after ordinary top-p filtering. Only the named smoothed KL metrics use the uniform mixture. Positive values favor correction; negative values are regressions. Green encodes positive point-estimate reductions, not significance. Percentage reductions in different metrics do not rank objectives. Ordinary post-filter KL remains infinite and is excluded.

Policy	Excluded BF16 mass	Quantized mass outside BF16	Sampling temperature (T)	Predicted top-p
Native MXFP4	0.0417	0.0378	0.600	0.950
Forward KL	0.0049	0.0823	1.502	0.975
Reverse KL	0.0638	0.0335	0.471	0.936
JSD	0.0223	0.0462	0.849	0.962

Table 16: **Support and controls at temperature $\tau = 0.6$.** Probabilities use ordinary top-p with no uniform mixture; masses are fractions. Learned temperature and cutoff entries are averages of adaptive actions across prefixes and both fits, not constant settings. The mass columns measure opposite directions of support error; green encodes relative reductions from native, not significance or task quality.

	Before top-p	After top-p				
Policy	Forward KL	Smoothed forward KL	Smoothed reverse KL	JSD	TV	Infinite KL forward/reverse
Native	0.6343	1.3459	1.3924	0.0557	0.1044	90/87
Forward-KL training						
Global T	0.4359	1.0215	1.4350	0.0523	0.1035	67/105
Global T, p	0.4375	0.9927	1.4365	0.0522	0.1035	66/107
Learned T, fit 1	0.2476	0.5119	1.9637	0.0563	0.1269	23/250
Learned T, fit 2	0.2469	0.5115	1.9624	0.0563	0.1269	23/250
Learned T, p, fit 1	0.2474	0.4563	1.9474	0.0559	0.1263	20/279
Learned T, p, fit 2	0.2469	0.4263	1.9570	0.0560	0.1268	18/280
Reverse-KL training						
Global T	0.8513	1.5699	1.3768	0.0582	0.1052	104/74
Global T, p	0.8413	1.5977	1.3757	0.0583	0.1052	107/72
Learned T, p, fit 1	0.6639	1.6275	1.3707	0.0584	0.1051	108/71
Learned T, p, fit 2	0.6482	1.6287	1.3726	0.0585	0.1052	109/71
JSD training						
Global T	0.4461	1.0279	1.4321	0.0525	0.1035	68/105
Global T, p	0.4475	0.9956	1.4319	0.0523	0.1034	67/105
Learned T, p, fit 1	0.3391	0.7889	1.5038	0.0507	0.1052	45/141
Learned T, p, fit 2	0.3384	0.7889	1.5048	0.0507	0.1053	45/142

Table 17: **Matched controls at temperature (τ) 0.3.** Divergences are in nats; TV is a fraction. All metrics after top-p use ordinary filtering; only smoothed columns mix both distributions with 10^{-4} uniform probability. The last column counts infinite ordinary-KL rows out of 768, separately by direction. Fits 1 and 2 use seeds 1729 and 3221. Learned actions vary by prefix; global actions are constant.

	Before top-p	After top-p				
Policy	Forward KL	Smoothed forward KL	Smoothed reverse KL	JSD	TV	Infinite KL forward/reverse
Native	0.3163	0.9117	0.8503	0.0430	0.0997	97/81
Forward-KL training						
Global T	0.2277	0.7124	1.0038	0.0419	0.1012	—
Global T, p	0.2291	0.6885	1.0159	0.0420	0.1016	—
Learned T, fit 1	0.1926	0.2961	1.7443	0.0519	0.1359	17/322
Learned T, fit 2	0.1922	0.2959	1.7420	0.0519	0.1358	17/320
Learned T, p, fit 1	0.1828	0.2720	1.5920	0.0490	0.1287	13/340
Learned T, p, fit 2	0.1826	0.2719	1.5911	0.0490	0.1287	13/341
Reverse-KL training						
Global T	0.4423	1.1403	0.8066	0.0469	0.1033	—
Global T, p	0.4408	1.2890	0.8073	0.0481	0.1041	—
Learned T, p, fit 1	0.4688	1.3265	0.8003	0.0488	0.1048	137/51
Learned T, p, fit 2	0.4691	1.3266	0.8004	0.0488	0.1048	137/51
JSD training						
Global T	0.2423	0.7325	0.9568	0.0417	0.0999	—
Global T, p	0.2505	0.6941	0.9598	0.0417	0.1002	—
Learned T, p, fit 1	0.2183	0.5579	0.9762	0.0409	0.1006	57/155
Learned T, p, fit 2	0.2181	0.5580	0.9767	0.0409	0.1007	57/155

Table 18: **Matched controls at temperature (τ) 0.6.** Conventions follow Table 17. The reused global reports record infinite ordinary KL but do not give integer infinite-row counts in the assembled table; dashes in the last column mean unavailable counts, not zero support failures or missing divergence measurements.

	Before top-p	After top-p				
Policy	Forward KL	Smoothed forward KL	Smoothed reverse KL	JSD	TV	Infinite KL forward/reverse
Native	0.1903	0.7486	0.6576	0.0359	0.0964	127/110
Forward-KL training						
Global T	0.1502	0.4158	0.9149	0.0359	0.1018	68/209
Global T, p	0.1508	0.3984	0.9222	0.0359	0.1020	59/226
Learned T, fit 1	0.1421	0.2387	1.2051	0.0393	0.1151	38/299
Learned T, fit 2	0.1416	0.2369	1.1973	0.0391	0.1146	37/301
Learned T, p, fit 1	0.1371	0.2108	1.0860	0.0373	0.1090	30/318
Learned T, p, fit 2	0.1369	0.2103	1.0831	0.0372	0.1087	29/320
Reverse-KL training						
Global T	0.2718	1.0296	0.5756	0.0397	0.1018	171/61
Global T, p	0.2698	1.0707	0.5706	0.0400	0.1020	176/53
Learned T, p, fit 1	0.3059	1.1957	0.5619	0.0422	0.1054	189/44
Learned T, p, fit 2	0.3059	1.1986	0.5614	0.0423	0.1054	189/43
JSD training						
Global T	0.1690	0.5581	0.7251	0.0348	0.0965	101/150
Global T, p	0.1708	0.5395	0.7277	0.0348	0.0966	96/157
Learned T, p, fit 1	0.1666	0.4941	0.7321	0.0344	0.0965	90/166
Learned T, p, fit 2	0.1666	0.4942	0.7324	0.0344	0.0965	90/166

Table 19: **Matched controls at temperature (τ) 1.0.** Conventions follow Table 17. The large forward-KL reductions coexist with worse reverse KL and TV; ordinary post-filter KL is still infinite. Rounding can hide small differences, so comparisons use the original unrounded values.

D.3 GEMMA FIDELITY AND EXPLORATORY DOWNSTREAM RESULTS

We also fit TempQuant for Gemma 4 31B, using the instruction-tuned BF16 model and its released MXFP4 counterpart, with BF16 KV storage.[1] This is a fixed-reference study at temperature 0.6 and top-p 0.95, not a three-temperature replication. We fit a Gemma-specific normalizer, two global controls, and two forward-KL controllers using 1,536 training prefixes; evaluation uses 512 reused development prefixes from 64 questions. The 448 source responses come from BF16 and native quantization with a 2,048-token limit. Both learned fits use the terminal 120-update checkpoint. Forward KL is the documented fallback, not a selected empirical winner.

	Before top-p	After top-p			
Policy	Forward KL	Smoothed forward KL	Smoothed reverse KL	JSD	TV
Native MXFP4	0.0744	0.3391	0.3175	0.0141	0.0314
Global T	0.0568	0.2464	0.3739	0.0141	0.0329
Global T, p	0.0571	0.1944	0.3738	0.0137	0.0327
TempQuant, fit 1	0.0549	0.0885	0.4693	0.0144	0.0364
TempQuant, fit 2	0.0546	0.0882	0.4683	0.0144	0.0364

Table 20: **Gemma 4 31B development fidelity.** Nats except TV. Native filtering is used; smoothed columns apply the same 10^{-4} mixture to both distributions. Both fits are retained, with no seed selection or paired confidence interval. These are fixed-prefix measurements, not downstream benchmark scores.

Both fits reduce smoothed forward KL by about 74% from native and 55% from the joint global. Before filtering, the reduction from native is about 26%, with only about 4% beyond the joint global. JSD worsens by 1.7–2.0%, TV by 15.7–16.0%, and smoothed reverse KL by 47.5–47.8%. Ordinary post-filter forward KL remains infinite: support failures occur on 18 of 512 native rows, 10 joint-global rows, and three rows for each learned fit. The BF16 target is very sharp, with mean post-filter entropy 0.0347 nats. These results do not establish downstream gains or broad generalization.

Exploratory downstream pilot. Gemma's separate pilot uses the first 32 canonical questions from each of MATH-500 and GPQA, with one response per question and deployed policy. The six policies are BF16, native MXFP4, the two fitted globals, and both forward-KL TempQuant fits. Temperature is 0.6, the baseline top-p is 0.95, and the output limit is 32,768 tokens. Unlike the Qwen reasoning evaluations, this pilot disables thinking mode. All 384 scores are known, and every response ended naturally. Table 21 reports this exploratory comparison, not a full benchmark replication.

	GPQA pilot		MATH pilot	
Policy	Correct / 32	Length (tokens)	Correct / 32	Length (tokens)
BF16	25/32	720.8	31/32	620.0
Native MXFP4	22/32	742.6	30/32	675.6
Global temperature	20/32	742.4	31/32	606.2
Global temperature/top-p	22/32	747.2	31/32	658.5
TempQuant, fit 1	21/32	761.2	30/32	686.0
TempQuant, fit 2	22/32	789.3	30/32	674.4

Table 21: **Gemma 4 31B exploratory downstream pilot.** Each task uses the same 32 questions and one original trial per policy, with no unknown scores or capped responses. Length includes terminal EOS. Both learned fits are retained. These are descriptive counts, without a significance test or a full-benchmark generalization claim.

Neither learned fit improves native accuracy on either task: MATH matches native in both fits, while GPQA matches in one fit and loses one correct answer in the other. One answer changes accuracy by 3.125 percentage points. These counts do not establish equivalence or select a fitting seed.

[1] Model releases: `https://huggingface.co/google/gemma-4-31B-it` and `https://huggingface.co/amd/gemma-4-31B-it-MXFP4`.

D.4 GSM8K ACCURACY AND RESPONSE LENGTH

The complete comparisons at 0.3, 0.6, and 1.0 each contain all 1,319 test questions, four trials per deployed policy, and both learned fits. All 42,208 scores per temperature are known. Table 22 retains the individual fits and nominal paired differences; no learned accuracy difference from native is resolved. This does not prove equivalence. All nine learned-minus-native intervals include zero. An equal-temperature pooled interval has not been computed. The 0.6 BF16/native rows reuse compatible earlier generation; the learned rows use the current controllers.

Policy	Fit 1 (%)	Fit 2 (%)	Mean − native [95% CI] (pp)
Temperature (τ) = 0.3			
BF16	94.47		—
Native MXFP4	93.67		—
Forward KL	93.65	93.84	$+0.08$ $[-0.33, +0.48]$
Reverse KL	93.54	93.48	-0.16 $[-0.56, +0.24]$
JSD	93.61	93.50	-0.11 $[-0.52, +0.27]$
Temperature (τ) = 0.6			
BF16	94.47		—
Native MXFP4	93.27		—
Forward KL	93.76	93.57	$+0.40$ $[-0.12, +0.91]$
Reverse KL	93.18	93.18	-0.09 $[-0.59, +0.41]$
JSD	93.52	93.80	$+0.39$ $[-0.20, +0.98]$
Temperature (τ) = 1.0			
BF16	95.02		—
Native MXFP4	93.65		—
Forward KL	93.99	93.82	$+0.26$ $[-0.15, +0.67]$
Reverse KL	93.37	93.59	-0.17 $[-0.57, +0.23]$
JSD	93.78	93.97	$+0.23$ $[-0.17, +0.63]$

Table 22: **GSM8K sampled accuracy.** Fit 1 is seed 1729; fit 2 is 3221. Differences average both fits within each question. Intervals use 10,000 paired whole-question bootstrap draws and condition on the fixed trained controllers; they are not familywise-adjusted. Two decimals retain the small effects. All intervals include zero.

	$\tau = 0.3$		$\tau = 0.6$		$\tau = 1.0$	
Policy	Length (K)	Caps	Length (K)	Caps	Length (K)	Caps
BF16	2.386	19	2.301	3	2.306	0
Native MXFP4	2.371	3	2.371	1	2.333	0
Forward KL, fit 1	2.335	0	2.381	0	2.388	0
Forward KL, fit 2	2.330	0	2.379	0	2.387	0
Reverse KL, fit 1	2.389	3	2.380	2	2.348	0
Reverse KL, fit 2	2.377	2	2.376	1	2.349	0
JSD, fit 1	2.372	0	2.357	0	2.342	0
JSD, fit 2	2.372	1	2.363	0	2.324	0

Table 23: **GSM8K length and output caps.** Mean generated length in thousands of tokens at each requested temperature; cap counts have denominator 5,276 in every cell. Caps remain in accuracy and length summaries. Historical 0.6 baseline completion-kind and string-stop labels are unavailable, although correctness, lengths, and cap indicators are known. These measurements do not establish looping.

D.5 MATH-500 ACCURACY AND RESPONSE LENGTH

The full 0.3 comparison includes all 500 questions, four trials per deployed policy, and both fits for each objective. All 16,000 grade records are present, with 15,904 known outcomes and 96 unknowns. The same three reference answers failed parsing or timed out in every policy and trial. These unknowns are scoring limitations, not generation failures or caps. Table 24 retains the full denominator, counting unknowns as incorrect for the lower bound and correct for the upper bound. These are missing-grade bounds, not confidence intervals. The main table uses their midpoints,

assigning half credit to every unknown outcome; this is a reporting convention, not fully observed accuracy. It is distinct from accuracy on the 497 gradeable questions. Table 1 uses the same midpoint convention at every temperature, including in its five-benchmark average. Shading uses unrounded gains over native, with separate scales for accuracy and ROUGE; it leaves regressions uncolored.

Policy	Correct / known	Accuracy bounds (%)	Length (K)	Caps / responses
BF16	1902/1988	95.100–95.700	5.375	20/2000
Native MXFP4	1833/1988	91.650–92.250	5.872	21/2000
Forward KL, fit 1	1834/1988	91.700–92.300	5.807	13/2000
Forward KL, fit 2	1853/1988	92.650–93.250	5.770	11/2000
Reverse KL, fit 1	1838/1988	91.900–92.500	5.745	22/2000
Reverse KL, fit 2	1836/1988	91.800–92.400	5.700	23/2000
JSD, fit 1	1846/1988	92.300–92.900	5.609	6/2000
JSD, fit 2	1850/1988	92.500–93.100	5.690	17/2000

Table 24: **Full MATH-500 at temperature** 0.3**.** Each deployed policy has 2,000 responses, including 12 unknown scores. Accuracy bounds use all 2,000 responses; correct/known counts do not define a replacement primary metric. Length and cap counts include every response. The main table averages both learned fits. No significance test is reported.

The learned two-fit ranges are 92.175–92.775% for forward KL, 91.850–92.450% for reverse KL, and 92.400–93.000% for JSD. All remain below BF16's lower bound. These descriptive ranges do not establish statistical significance or select a training objective.

At 0.6, 488 of the original 2,000 native responses were lost before collection. We therefore report the two original trials whose responses are fully available for every policy and all 500 questions, retaining both learned fits. This retrospective choice is based on availability, not correctness; the losses cannot be assumed random. The same three reference answers are ungradable. Table 25 retains all 1,000 responses per policy for bounds, length, and caps. It does not represent the originally planned four-trial comparison. No paired confidence interval or significance test is reported.

Policy	Correct / known	Accuracy bounds (%)	Length (K)	Caps / responses
BF16	950/994	95.00–95.60	5.187	2/1000
Native MXFP4	922/994	92.20–92.80	5.815	9/1000
Forward KL, fit 1	920/994	92.00–92.60	5.936	7/1000
Forward KL, fit 2	931/994	93.10–93.70	5.804	6/1000
Reverse KL, fit 1	927/994	92.70–93.30	5.753	12/1000
Reverse KL, fit 2	923/994	92.30–92.90	5.729	9/1000
JSD, fit 1	929/994	92.90–93.50	5.745	10/1000
JSD, fit 2	930/994	93.00–93.60	5.849	6/1000

Table 25: **Matched two-trial MATH-500 at temperature** 0.6**.** Each policy has 994 known grades and six unknowns from three reference answers. Bounds include the unknown grades, not sampling uncertainty. Length and caps use all 1,000 available responses per policy.

At 1.0, all 16,000 responses are scored, again with 15,904 known outcomes and 96 unknown grades from the same three reference answers. Both fits remain included (Table 26). Across temperatures, the learned recipe averages exceed native but remain below BF16. JSD has the highest quantized estimate at the two lower temperatures; forward KL has the highest at 1.0. These are descriptive rankings without paired confidence intervals, and they do not establish a winning training loss.

D.6 GPQA DECODING CONTROLS

These completed panels compare forward-KL TempQuant with a fitted global temperature at 0.3 and an independently trained temperature-only controller at 0.6. Every deployed policy uses the same 198 questions, eight trials, and original scoring procedure. Table 27 retains both fits and completion counts.

At 0.3, joint TempQuant differs from native by -1.4 percentage points ($[-3.9, +1.2]$) and from the global temperature by $+0.6$ points ($[-1.9, +3.1]$). At 0.6, it differs from the separately trained temperature-only controller by $+0.5$ points ($[-1.3, +2.2]$). The incremental benefit is unresolved

Policy	Correct / known	Accuracy bounds (%)	Length (K)	Caps / responses
BF16	1905/1988	95.25–95.85	5.340	8/2000
Native MXFP4	1845/1988	92.25–92.85	5.939	18/2000
Forward KL, fit 1	1857/1988	92.85–93.45	5.890	12/2000
Forward KL, fit 2	1859/1988	92.95–93.55	5.892	7/2000
Reverse KL, fit 1	1858/1988	92.90–93.50	5.731	17/2000
Reverse KL, fit 2	1856/1988	92.80–93.40	5.754	18/2000
JSD, fit 1	1856/1988	92.80–93.40	5.810	13/2000
JSD, fit 2	1839/1988	91.95–92.55	5.863	16/2000

Table 26: **MATH-500 at temperature** 1.0**.** Four trials per question; bounds, length, and caps include all 2,000 responses per policy. Bounds reflect unknown grades, not sampling uncertainty.

Policy	Accuracy (%)	Length (K)	Caps / responses
Temperature (τ) = 0.3			
BF16	39.8	9.877	113/1,584
Native	41.5	7.582	22/1,584
Global temperature	39.5	7.401	15/1,584
Joint, fit 1	40.5	7.444	0/1,584
Joint, fit 2	39.7	7.357	0/1,584
Temperature (τ) = 0.6			
BF16	41.5	8.592	11/1,584
Native	39.3	7.423	6/1,584
Temperature only, fit 1	42.2	7.707	2/1,584
Temperature only, fit 2	41.0	7.726	0/1,584
Joint, fit 1	42.6	7.560	1/1,584
Joint, fit 2	41.5	7.522	1/1,584

Table 27: **Completed GPQA control panels.** All scores and lengths are known. Fits 1 and 2 use seeds 1729 and 3221. The 0.3 global control samples at fitted temperature 0.4401 and cutoff 0.95; requested temperature labels refer to the BF16 target. The temperature-only ablation also fixes its cutoff at 0.95. These are the two reported downstream control comparisons.

in both comparisons. Intervals use 10,000 paired whole-question resamples and average both fits within each question. These control contrasts are nominal, without multiplicity adjustment. Fewer caps alone does not establish a better answer or explain why accuracy changes.

D.7 XSum summary quality

All 24,576 responses from the 128-article XSum validation subset were scored. Table 28 gives per-fit ROUGE-2 and paired differences from native quantization; Table 1 reports the pooled scores.

The paired-swap sensitivity also gives no nominal $p < .05$ contrast. XSum quality does not select a training objective or fit seed.

Training loss	Seed 1729	Seed 3221	Difference [95% CI] (points)
Temperature (τ) = 0.3			
Forward KL	5.58	5.54	-0.06 $[-0.36, +0.24]$
Reverse KL	5.59	5.58	-0.04 $[-0.11, +0.03]$
JSD	5.62	5.58	-0.02 $[-0.21, +0.17]$
Temperature (τ) = 0.6			
Forward KL	5.54	5.52	-0.06 $[-0.36, +0.23]$
Reverse KL	5.68	5.70	$+0.10$ $[-0.06, +0.26]$
JSD	5.64	5.66	$+0.05$ $[-0.14, +0.25]$
Temperature (τ) = 1.0			
Forward KL	5.60	5.55	-0.07 $[-0.32, +0.16]$
Reverse KL	5.70	5.67	$+0.03$ $[-0.14, +0.21]$
JSD	5.62	5.62	-0.03 $[-0.15, +0.10]$
Equal-temperature pooled			
Forward KL	Equal temperature weights		-0.07 $[-0.28, +0.14]$
Reverse KL	Equal temperature weights		$+0.03$ $[-0.06, +0.12]$
JSD	Equal temperature weights		$+0.003$ $[-0.108, +0.112]$

Table 28: **XSum quality: ROUGE-2 vs. human reference.** Fixed 128-article validation subset, ROUGE-2 F1 on the 0–100 scale. Differences are the two-fit mean minus native MXFP4, with nominal paired 95% intervals (Appendix C.3). All intervals include zero; none establishes a gain or equivalence. The small pooled JSD difference retains an extra decimal. Green marks favorable point estimates, not significance.

D.8 XSUM SIMILARITY TO BF16 AND RESPONSE BEHAVIOR

The same responses supply BLEU against BF16 (Tables 3 and 29) and within-policy diversity, length, and repetition (Table 30). Metric definitions appear in Appendix C.3.

Better distributional fidelity does not uniformly improve output-level agreement. Forward-KL training lowers direct BF16 BLEU at every temperature and moves self-BLEU farther from BF16, while reverse KL improves some of these measures. These lexical measurements do not establish semantic agreement.

Training loss	Seed 1729	Seed 3221	Difference [95% CI] (points)
Temperature (τ) = 0.3			
Forward KL	44.43	44.40	-1.28 $[-2.30, -0.17]$
Reverse KL	45.65	45.68	-0.02 $[-0.28, +0.26]$
JSD	45.85	45.86	$+0.16$ $[-0.39, +0.77]$
Temperature (τ) = 0.6			
Forward KL	46.13	46.19	-2.56 $[-3.62, -1.53]$
Reverse KL	49.01	48.80	$+0.18$ $[-0.30, +0.66]$
JSD	48.13	48.13	-0.59 $[-1.12, +0.08]$
Temperature (τ) = 1.0			
Forward KL	48.17	48.23	-2.37 $[-3.22, -1.51]$
Reverse KL	51.49	51.55	$+0.95$ $[+0.28, +1.71]$
JSD	49.72	49.91	-0.75 $[-1.28, -0.21]$
Equal-temperature pooled			
Forward KL	Equal temperature weights		-2.07 $[-2.87, -1.25]$
Reverse KL	Equal temperature weights		$+0.37$ $[+0.05, +0.70]$
JSD	Equal temperature weights		-0.39 $[-0.71, -0.01]$

Table 29: **XSum reference similarity: BLEU vs. BF16.** Corpus BLEU points (0–100). Differences are the two-fit mean minus native MXFP4, with nominal paired 95% intervals. The two fits remain separate policies, not a concatenated reference set. Positive changes here move toward the larger BF16/BF16 statistic; this is lexical similarity, not semantic agreement or human-reference quality. Green represents favorable point-estimate changes, not statistical significance; negative differences remain uncolored.

No XSum responses are missing, empty, malformed, or excluded. Each policy and fit has 1,024 responses per temperature. BF16 has one capped response at $\tau = .3$ ($.1\%$), affecting one article; every other group has zero caps. These short outputs provide no evidence that the lexical changes come from correcting loops.

Policy	Self-BLEU	Absolute deviation	Mean length (tokens)	Repeated 4-grams (10^{-4})
Temperature (τ) = 0.3				
BF16	95.44	0.00	43.96	9.94
Native MXFP4	86.79	9.17	42.31	0.60
TempQuant, forward KL	73.03	22.41	42.42	1.91
TempQuant, reverse KL	88.53	7.86	42.28	0.60
TempQuant, JSD	81.80	13.88	42.32	0.41
Temperature (τ) = 0.6				
BF16	90.17	0.00	43.55	0.45
Native MXFP4	81.74	9.77	42.29	0.38
TempQuant, forward KL	68.41	21.86	42.58	0.50
TempQuant, reverse KL	84.72	7.93	42.29	0.52
TempQuant, JSD	77.56	13.00	42.06	0.69
Temperature (τ) = 1.0				
BF16	83.02	0.00	43.77	6.80
Native MXFP4	74.28	10.34	42.52	1.24
TempQuant, forward KL	67.16	16.04	42.37	1.35
TempQuant, reverse KL	80.41	8.10	42.10	1.55
TempQuant, JSD	71.98	12.19	42.53	1.03

Temperature (τ)	Forward KL	Reverse KL	JSD
0.3	$+13.24$ $[+11.86, +14.67]$	-1.31 $[-1.96, -0.69]$	$+4.71$ $[+3.80, +5.63]$
0.6	$+12.09$ $[+10.73, +13.53]$	-1.85 $[-2.75, -0.97]$	$+3.23$ $[+2.25, +4.19]$
1.0	$+5.71$ $[+4.57, +6.84]$	-2.24 $[-3.37, -1.08]$	$+1.86$ $[+1.06, +2.66]$

Table 30: **XSum diversity, length, and repetition.** Self-BLEU and per-article absolute deviation from BF16 use the 0–100 scale. The lower panel gives learned-minus-native changes in that deviation [nominal paired 95% interval]; negative is closer to BF16. Green marks reductions in this reference-deviation measure, not significance. Diversity, length, and repetition themselves are uncolored: smaller or larger is not automatically better. Repeated-fourgram fractions are multiplied by 10^4 for readability, not percentages or loop rates. Learned rows average both fit statistics.

D.9 GPQA ACCURACY, RESPONSE LENGTH, AND CAPS

GPQA is complete at all three temperatures: 198 questions and eight trials give 1,584 scored responses per policy and fit, with no missing outcomes. Table 31 reports both fits and paired uncertainty; Table 32 reports completion at $\tau = .3$.

Training loss	Seed 1729	Seed 3221	Difference [95% CI] (pp)
Temperature (τ) = 0.3			
Forward KL	$642/1584$	$629/1584$	-1.4 $[-3.9, +1.2]$
Reverse KL	$622/1584$	$627/1584$	-2.1 $[-4.5, +0.3]$
JSD	$638/1584$	$614/1584$	-2.0 $[-4.3, +0.4]$
Temperature (τ) = 0.6			
Forward KL	$675/1584$	$657/1584$	$+2.8$ $[+0.5, +5.1]$
Reverse KL	$628/1584$	$622/1584$	$+0.2$ $[-2.1, +2.5]$
JSD	$629/1584$	$661/1584$	$+1.5$ $[-0.8, +3.7]$
Temperature (τ) = 1.0			
Forward KL	$643/1584$	$654/1584$	$+1.4$ $[-0.9, +3.7]$
Reverse KL	$615/1584$	$630/1584$	-0.3 $[-2.9, +2.3]$
JSD	$637/1584$	$660/1584$	$+1.4$ $[-0.9, +3.6]$

Table 31: **GPQA accuracy and paired uncertainty.** Seed columns retain exact counts of correct/all responses. Differences are the equal-fit mean accuracy minus native MXFP4, with nominal paired 95% intervals in percentage points. BF16/native counts are $630/1584$ and $657/1584$ at .3, $658/1584$ and $622/1584$ at .6, and $677/1584$ and $627/1584$ at 1.0. Green represents favorable point estimates, not study-wide significance. No objective or seed is selected.

Policy / fit seed	Capped / 1,584	Affected / 198	Mean length (K tokens)
BF16	113	57	9.88
Native MXFP4	22	15	7.58
Forward KL / 1729	0	0	7.44
Forward KL / 3221	0	0	7.36
Reverse KL / 1729	47	29	8.04
Reverse KL / 3221	38	27	7.83
JSD / 1729	4	3	7.23
JSD / 3221	12	7	7.33

Table 32: **GPQA completion at temperature $\tau = 0.3$.** Output cap: 32,768 tokens. All responses remain in accuracy. Affected questions have at least one capped response among eight trials. Length includes all raw generated tokens. No favorability shading is used: a cap is not necessarily a loop or an incorrect answer.

Equal-fit cap-rate differences from native are -1.4 percentage points for forward KL ($[-2.2, -.7]$), $+1.3$ for reverse KL ($[+.5, +2.1]$), and $-.9$ for JSD ($[-1.5, -.3]$). BF16 has more capped responses than native quantization and every learned fit at .3. This does not support a simple account in which quantization uniformly increases looping; cap counts alone do not identify loops.

At $\tau = .6$, cap counts are BF16 $11/1584$, native $6/1584$, forward KL $2/3168$, reverse KL $43/3168$, and JSD $8/3168$, with learned counts pooled across both fits. Mean lengths are 8.59, 7.42, 7.54, 7.61, and 7.30 thousand tokens, respectively. Forward KL has fewer caps but slightly longer average outputs than native quantization; these summaries do not identify why accuracy changes.

At $\tau = 1.0$, cap counts are BF16 $5/1584$, native $0/1584$, forward KL $0/3168$, reverse KL $4/3168$, and JSD $2/3168$, with learned counts pooled across the two fits. Mean lengths are 8.561, 7.372, 7.604, 7.174, and 7.419 thousand tokens, respectively. The reverse-KL groups have known cap flags and token lengths but incomplete string-stop subtype metadata.

Reused baseline trials. At .6, GPQA and AIME baselines combine two earlier trials with six new trials per question, without outcome selection or regrading. Reuse checks do not establish numerical identity between the execution paths. Grades, lengths, and cap flags are known; earlier timing and some stop subtypes are unavailable. Prior evaluation exposure also limits independence.

D.10 AIME ACCURACY, RESPONSE LENGTH, AND CAPS

All 7,680 responses per temperature have known grades, lengths, and cap flags. Table 33 gives per-fit accuracy and paired uncertainty for the 120 AIME questions; Table 34 gives completion. Capped responses remain in scoring.

Training loss	Seed 1729	Seed 3221	Difference [95% CI] (pp)
Temperature (τ) = 0.3			
Forward KL	$464/960$	$456/960$	$+4.7$ $[+2.2, +7.1]$
Reverse KL	$421/960$	$426/960$	$+0.9$ $[-1.6, +3.4]$
JSD	$451/960$	$433/960$	$+2.8$ $[+0.3, +5.4]$
Temperature (τ) = 0.6			
Forward KL	$455/960$	$446/960$	$+1.1$ $[-1.1, +3.4]$
Reverse KL	$429/960$	$426/960$	-1.3 $[-3.7, +1.0]$
JSD	$461/960$	$461/960$	$+2.2$ $[+0.1, +4.3]$
Temperature (τ) = 1.0			
Forward KL	$443/960$	$451/960$	-1.6 $[-4.3, +1.2]$
Reverse KL	$467/960$	$469/960$	$+0.6$ $[-2.0, +3.3]$
JSD	$464/960$	$461/960$	$+0.05$ $[-2.50, +2.71]$

Table 33: **AIME accuracy and paired uncertainty.** Exact correct/all response counts by fit; differences are the equal-fit mean minus native, with nominal paired 95% intervals in percentage points (Appendix C.3). BF16/native counts are $630/960$ and $415/960$ at .3, $655/960$ and $440/960$ at .6, and $665/960$ and $462/960$ at 1.0. The small JSD difference at 1.0 retains an extra decimal. Intervals are not adjusted for multiple comparisons.

The gains at some settings coexist with negative estimates elsewhere and a large gap to BF16. In particular, JSD's 2.2-point gain over native at .6 still leaves a 20.2-point BF16 gap. Neither these differences nor the cap counts identify a causal mechanism or select a training objective.

	Capped / attempts			Mean length (K tokens)		
Temperature (τ)	0.3	0.6	1.0	0.3	0.6	1.0
BF16	$135/960$	$90/960$	$80/960$	17.64	16.91	17.06
Native MXFP4	$111/960$	$104/960$	$77/960$	18.40	18.24	18.67
Forward KL	$127/1920$	$140/1920$	$142/1920$	18.04	18.44	18.73
Reverse KL	$227/1920$	$272/1920$	$208/1920$	18.19	18.72	18.28
JSD	$166/1920$	$177/1920$	$154/1920$	17.92	18.25	18.48

Table 34: **AIME completion across temperatures.** Learned counts pool both fits; lengths average their means. Output cap: 32,768 tokens. Raw token length includes generated EOS and excludes the prompt. Caps are budget exhaustion, not identified loops.

Accuracy averaged across temperatures. Table 35 averages the three temperatures equally, with all trials and temperatures of each question resampled together. It summarizes a different comparison from any individual temperature.

	GPQA		AIME	
Policy	Accuracy (%)	Difference [95% CI] (pp)	Accuracy (%)	Difference [95% CI] (pp)
BF16	41.4	–	67.7	–
Native MXFP4	40.1	Reference	45.7	Reference
Forward KL	41.0	$+0.9$ $[-0.6, +2.5]$	47.1	$+1.4$ $[-0.2, +2.9]$
Reverse KL	39.4	-0.7 $[-2.3, +0.9]$	45.8	$+0.07$ $[-1.46, +1.56]$
JSD	40.4	$+0.3$ $[-1.2, +1.7]$	47.4	$+1.7$ $[+0.3, +3.0]$

Table 35: **Accuracy averaged equally over temperatures** $.3, .6, 1.0$**.** Learned rows average both fits. Differences are relative to native, with nominal paired whole-question 95% intervals. The 198 GPQA and 120 AIME questions are the resampling units, not their repeated responses. The small AIME reverse-KL difference retains an extra decimal. Intervals are not adjusted for multiple comparisons.

D.11 ENTROPY ON THE SAME PREFIXES

This diagnostic compares both models on the same prefixes, holding context fixed. It is distinct from the own-generation experiment in Figure 3.

Each of the 198 GPQA-Diamond questions supplies one BF16 and one quantized response generated at $(.6, .95)$. Both models are replayed on both response sources: 1,689,983 positions from BF16 and 1,498,371 from quantization. All positions are included, with three capped BF16 responses and no missing data. Replay at .3 and 1.0 reuses these .6 histories; it does not measure independent generation at those temperatures.

Means average positions within a response, then questions equally. Table 36 uses 10,000 paired question-bootstrap draws that retain both models, response sources, filtering stages, and temperatures. The 98.75% intervals adjust for four prespecified contrasts by Bonferroni and condition on the single trial. Quantization raises mean entropy on BF16 histories and lowers it on quantized histories. Token-weighted means have the same signs. These are conditional-entropy findings, not evidence of a TempQuant or accuracy gain.

History source	Stage	BF16	Quantized	Difference [98.75% CI]
BF16	Before top-p	0.2395	0.2477	$+0.0082$ $[+0.0062, +0.0102]$
	After top-p	0.1984	0.2047	$+0.0063$ $[+0.0046, +0.0079]$
Quantized	Before top-p	0.2659	0.2528	-0.0131 $[-0.0156, -0.0107]$
	After top-p	0.2203	0.2089	-0.0114 $[-0.0135, -0.0094]$

Table 36: **Paired GPQA entropy at temperature $\tau = 0.6$.** All entropy values and quantized-minus-BF16 differences are in nats. Each row compares both models on the same prefixes; history sources are not pooled. Post-filter rows use ordinary top-$p = .95$, without uniform mixing. Intervals cover the prespecified four-contrast family and condition on one saved trial. No direction of entropy change is labeled a performance gain.

Figure 6 smooths 2,048 bins over $[0, \log(151936)]$ with a Gaussian bandwidth of about .0932 nats, reflected at both boundaries and normalized. Exact-zero mass is kept separate; it is zero here, while near-zero values contribute to the peak. Means and intervals use unsmoothed values. Replay proceeds incrementally from the original prompt: processing full prefixes at once gave different numerical results, so execution-path invariance is not established.

Entropy-equivalent temperature on fixed prefixes (not estimated). For the same prefixes and normalized question-balanced position weights w_i, the proposed mapping solves

$$\sum_i w_i H\left(\mathrm{softmax}\left(\frac{u_i}{T_{\mathrm{eq}}(\tau)}\right)\right) = \sum_i w_i H\left(\mathrm{softmax}\left(\frac{z_i}{\tau}\right)\right). \tag{10}$$

Each response source would have its own mapping. The three replay temperatures do not resolve it reliably, so no inverse is reported. This fixed-prefix quantity cannot supply the generation-level mapping below.

D.12 ENTROPY DURING INDEPENDENT GENERATION

Figure 3 compares each model along its own generations at deployed temperatures 0.3, 0.6, and 1.0. Both models use the same 198 GPQA questions, eight trials per question, prompts, top-$p = 0.95$, and 32,768-token output limit. Both use BF16 KV; the quantized model uses native MXFP4 weights and activations. Each saved response is replayed through its originating model, using incremental decoding. All 9,504 responses and 78,260,336 continuation positions are included. Of these, 396 responses at 0.6 reuse the existing own-model replay; the rest are newly replayed. No histories generated at 0.6 stand in for generation at another temperature.

The measurements are provisional because agreement with the historical live-generation execution path has not been fully validated. They are replay-derived next-token entropies, not entropy recorded during generation. The models follow different histories, so the comparison combines conditional prediction changes with changed trajectory visitation.

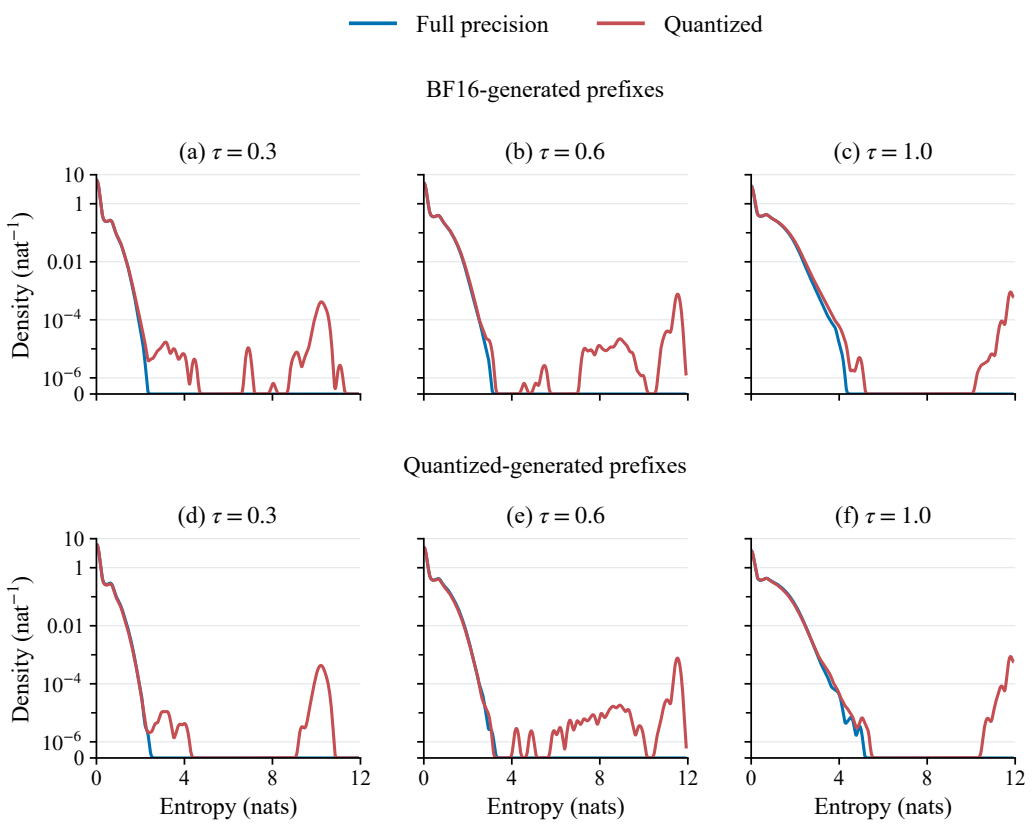

Figure 6: **Full-range GPQA entropy distributions before top-p.** Rows distinguish the source of the saved responses, not which model is evaluated: both models predict at every prefix in each row. Full precision is blue; quantization is solid red. Columns vary the diagnostic temperature on the same histories generated at .6. All curves retain question-balanced weights and the full support $[0, \log V]$; rare high-entropy quantized tails remain visible. The density axis is linear through 10^{-6} and logarithmic above it. Smoothing affects display only; the full-range normalization is retained. These are fixed-history diagnostics, not a deployed-temperature sweep of independent generation.

Entropy is measured over the entire next-token distribution, separately before and after ordinary top-p. Means average positions within responses, trials within questions, then questions equally. Distributions use the same weights; token-weighted means are a sensitivity. Capped responses remain included, with lengths and stop reasons reported. No responses are missing.

For either filtering stage s, let $\bar{H}_{M,s}(\tau)$ be this mean on model M's own generations at τ. The generation-level mapping solves

$$\bar{H}_{\mathrm{BF16},s}\left(T^{\mathrm{gen}}_{\mathrm{eq},s}(\tau)\right) = \bar{H}_{\mathrm{Q},s}(\tau). \tag{11}$$

Changing trajectories can make the BF16 curve nonmonotonic. The measured means increase across the three BF16 anchors, but a precise inverse is not resolved. Quantized mean entropy at 0.3 lies between the BF16 0.3 and 0.6 anchors; at 0.6, it lies between BF16 0.6 and 1.0. These intervals bracket observed entropy levels, not confidence intervals or evidence of a unique match between anchors. Quantized entropy at 1.0 exceeds the observed BF16 range. We do not extrapolate or infer exact temperatures from these sparse measurements. Equal mean entropy would not imply equal probabilities, rankings, answers, or exact logit scaling.

Uncertainty uses 10,000 paired whole-question bootstrap draws, retaining all models, temperatures, and trials together. The nominal 95% intervals condition on the eight saved trials and do not cover fresh-generation variability or multiple comparisons. Token positions are not independent experimental units. These own-trajectory intervals are recalculated; the shared-prefix intervals are not reused.

Temperature (τ)	Stage	BF16	MXFP4	Difference [95% CI]
0.3	Before top-p	0.1002	0.1101	+0.0098 [+0.0075, +0.0123]
	After top-p	0.0838	0.0923	+0.0085 [+0.0065, +0.0106]
0.6	Before top-p	0.2414	0.2497	+0.0083 [+0.0050, +0.0116]
	After top-p	0.2000	0.2065	+0.0065 [+0.0037, +0.0092]
1.0	Before top-p	0.4748	0.4916	+0.0168 [+0.0096, +0.0239]
	After top-p	0.3907	0.4018	+0.0111 [+0.0053, +0.0169]

Table 37: **Own-generation mean entropy (provisional).** Nats, question/trial-balanced. Differences are MXFP4 minus BF16 on each model's own responses. Intervals are nominal paired 95%, not familywise-adjusted. All six differences are positive; this direction is not a task-quality claim.

Model	Temperature (τ)	Positions	Caps	Token-weighted before top-p	Token-weighted after top-p
BF16	0.3	15,644,965	113	0.0821	0.0687
MXFP4	0.3	12,009,355	22	0.0998	0.0837
BF16	0.6	13,609,394	11	0.2291	0.1898
MXFP4	0.6	11,758,601	6	0.2369	0.1959
BF16	1.0	13,560,054	5	0.4628	0.3806
MXFP4	1.0	11,677,967	0	0.4797	0.3919

Table 38: **Entropy coverage and token-weighted means (provisional).** Each row contains 1,584 responses; caps remain included, and all other responses reached natural EOS. Entropies are in nats. Token weighting gives longer responses greater influence and is secondary to the question-balanced analysis. It gives the same direction of difference at every temperature.

BF16 reaches the cap more often than native MXFP4, especially at 0.3. Capped subsets have lower mean entropy than uncapped subsets in each condition where caps occur. These outcome-defined groups do not establish why the models differ, or show that quantization causes more looping.

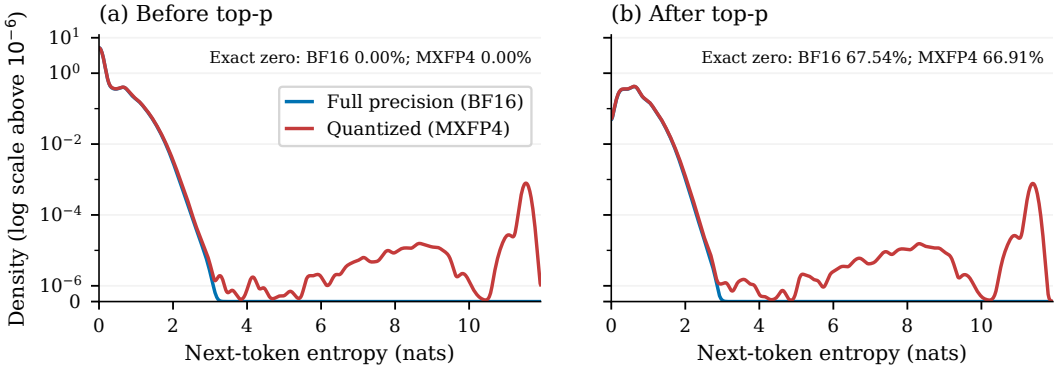

Figure 7: **Own-generation entropy distributions at temperature** 0.6 **(provisional).** Full precision is blue and native MXFP4 is solid red. The curves retain question/trial-balanced weights and the full entropy range; exact-zero mass is reported separately. Positive-entropy bins are smoothed for display, not for computing means or intervals. The vertical axis is linear through 10^{-6} and logarithmic above it.

The distribution display uses 2,048 saved bins over $[0, \log(151936)]$ and a reflected Gaussian bandwidth of about 0.0932 nats. Exact-zero mass is excluded from smoothing; positive density integrates to the remaining mass. Bin centers approximate the original values only for display.

D.13 Token probabilities on the same prefixes

Figures 8 and 9 average probabilities over 768 fixed development positions from 64 GSM8K and MATH training questions at $\tau = .6$. They use a separate population from the GPQA entropy study.

The left panels sort each model's tokens independently; the right panels use BF16's pre-filter token order for both models. The latter reveal token-identity disagreements that independent sorting can hide. The 512 displayed ranks are not renormalized. The probability axis is logarithmic above .00001% and linear below it to include zero; values near 10^{-18} can be subtraction residuals. These are fixed-prefix averages, not free-running generation results.

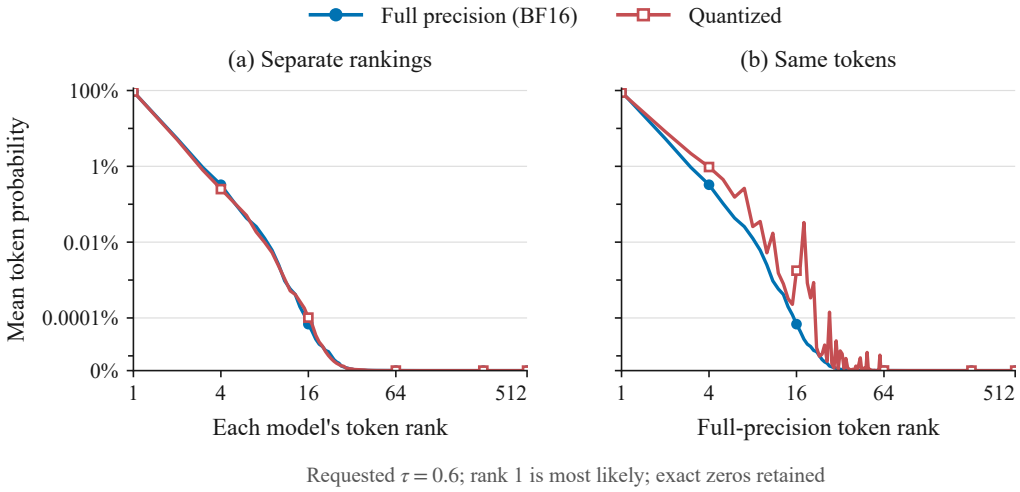

Figure 8: **Token probabilities before filtering.** Requested $\tau = .6$ on the same 768 development positions. Left: tokens are ranked separately within each model. Right: both curves refer to the same tokens, ordered by full-precision probability at each prefix. Each curve averages over positions. Logarithmic axes expose low-probability differences; the probability axis includes exact zero.

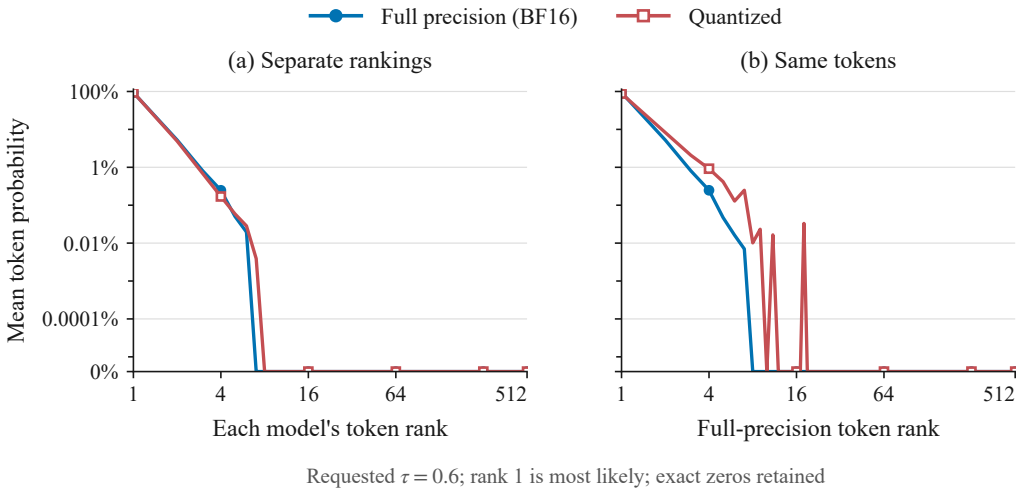

Figure 9: **Token probabilities after filtering.** Same prefixes, rank conventions, and requested setting as Figure 8, after ordinary top-p filtering at .95 for both models. Zero probability means a token was excluded or received zero numerical probability. No uniform mixture or displayed-rank renormalization is used.

E DIVERGENCE MEASURES AND ORACLE ANALYSIS

E.1 FORWARD KL, REVERSE KL, AND JSD

Forward KL weights disagreement by reference probabilities p; reverse KL weights it by candidate probabilities q. JSD compares both with their mixture. All three use natural logarithms. For $m = \frac{p+q}{2}$,

$$\mathrm{KL}(p\|q) = \sum_{i:p_i>0} p_i \log\left(\frac{p_i}{q_i}\right),$$
$$D_{\mathrm{JS}}(p,q) = \frac{1}{2}\,\mathrm{KL}(p\|m) + \frac{1}{2}\,\mathrm{KL}(q\|m). \tag{12}$$

Reverse KL exchanges p and q in the first expression. JSD is symmetric, lies between zero and $\log 2$ nats, and is zero exactly when $p = q$.

Synthetic distribution examples. Figure 10 fits a single Gaussian to a two-peak reference, following the example in Appendix A.7 of Agarwal et al. (2024) with our own distributions and an added JSD fit. The examples illustrate how the preferred approximation depends on the reference distribution; they do not predict which controller will work best.

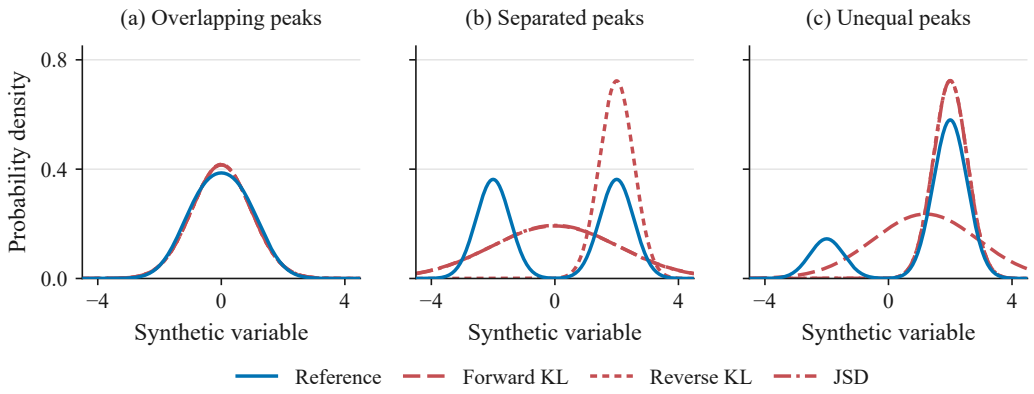

Figure 10: **KL and JSD approximations.** Original synthetic examples inspired by Agarwal et al. (2024), Appendix A.7. The blue curve is a two-Gaussian reference; red curves are single-Gaussian fits under the three objectives. Nearly coincident curves are intentional. The reference components have means $\pm.6$ and standard deviation $.75$ in (a), and means ± 2 and standard deviation $.55$ in (b,c). Mixture weights are equal in (a,b) and $.2/.8$ in (c). These are illustrative distributions, not model measurements.

Forward KL matches the reference mean and variance; the other fits use bounded numerical optimization. Exact synthetic inputs and optimization settings are retained with the figure sources.

Support mismatch. Forward KL is infinite when the candidate excludes a token with positive reference probability. Reverse KL is infinite for the opposite exclusion. JSD stays finite because its mixture covers both supports (Figure 11). Disjoint supports give JSD $\log 2$.

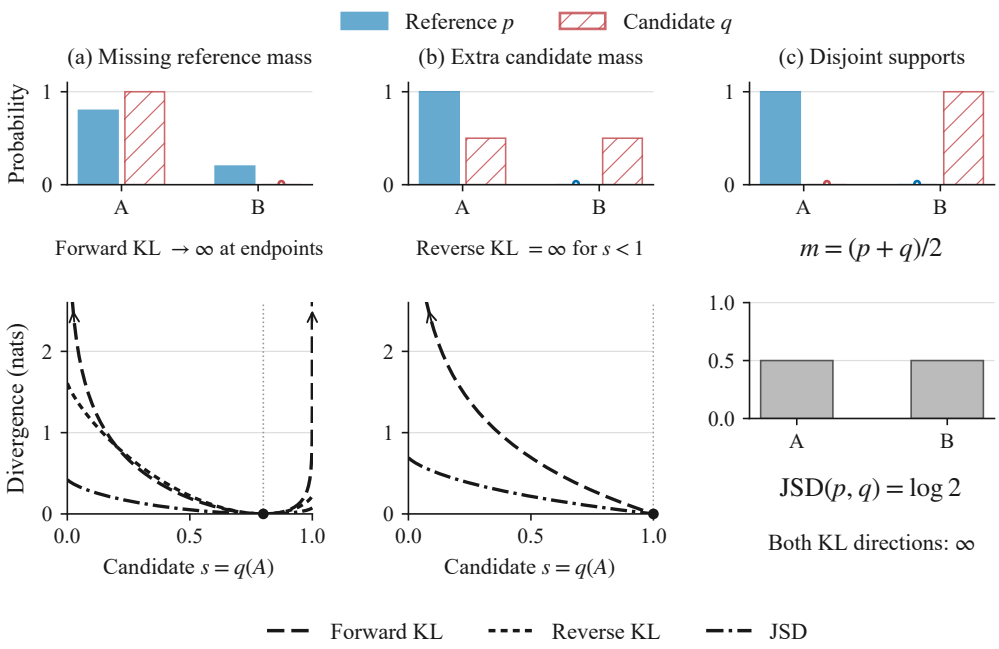

Figure 11: **KL under support mismatch.** The candidate probabilities are $(s, 1-s)$. The reference is $(.8, .2)$ in (a) and $(1, 0)$ in (b). Forward KL diverges at both endpoints in (a) and at $s = 0$ in (b); upward arrows mark curves continuing beyond the displayed range. Reverse KL in (b) is infinite for every $s < 1$ and has no finite curve there. All three divergences are zero at the black reference marker. Panel (c) shows disjoint supports, where both KL directions are infinite but JSD is $\log 2$. JSD remains finite without replacing zero probabilities.

E.2 KL WITH SMOOTHED PROBABILITIES

To keep KL finite after filtering, we mix each distribution with a small uniform component. For output dimension V, let $U_i = 1/V$ and define $P_\eta = (1-\eta)P + \eta U$ and $Q_\eta = (1-\eta)Q + \eta U$. The two discrepancies are

$$D_{F,\eta}(P,Q) = \mathrm{KL}(P_\eta \| Q_\eta), \qquad D_{R,\eta}(P,Q) = \mathrm{KL}(Q_\eta \| P_\eta). \tag{13}$$

The mixture weight is $\eta = 10^{-4}$. Training mixes the smooth student distribution; evaluation mixes the ordinary top-p distribution. BF16 is filtered before mixing in both cases. JSD and generation use no uniform mixture.

Training scales either KL loss by $1/4$ relative to the 0.01 variance penalty, matching JSD's local quadratic curvature before mixing. Reported divergences are unscaled; this does not make the objectives globally equivalent.

The mixture weight matters especially when support is missing. If $M = \sum_{i:Q_i=0} P_i > 0$, then as $\eta \to 0$,

$$D_{F,\eta}(P,Q) = M \log \frac{1}{\eta} + O(1). \tag{14}$$

Reverse KL has the same form with P and Q exchanged. We do not establish robustness to the mixture weight. Uniform mixing keeps KL finite; the separate smooth top-p approximation supplies filtering gradients.

E.3 TEMPERATURE-ONLY ORACLE

This oracle finds the best forward-KL match available by changing temperature at a fixed prefix. It requires the BF16 distribution. For a fixed target p and unfiltered quantized distribution $q_\beta = \mathrm{softmax}(\beta z)$, write $\beta = \frac{1}{T}$ and $\mu = \sum_i p_i z_i$. The forward-KL objective and its derivatives are

$$L(\beta) = \log \sum_i e^{\beta z_i} - \beta\mu - H(p), \qquad L'(\beta) = \mathbb{E}_{q_\beta}[z] - \mu, \qquad L''(\beta) = \mathrm{Var}_{q_\beta}(z). \tag{15}$$

Finite nonconstant logits make L strictly convex. A nonnegative derivative at the lower endpoint selects that endpoint; a nonpositive derivative at the upper endpoint selects the upper one. Otherwise the unique interior solution satisfies $\mathbb{E}_{q_\beta}[z] = \mathbb{E}_p[z]$.

Remaining KL after temperature correction. Let $A(\beta) = \log \sum_i e^{\beta z_i}$ and $q^* = q_{\beta^*}$. Then

$$\begin{aligned} \mathrm{KL}(p\|q_\beta) = \mathrm{KL}(p\|q^*) + \mathrm{KL}(q^*\|q_\beta) \\ + (\beta - \beta^*)\{A'(\beta^*) - \mathbb{E}_p[z]\}. \end{aligned} \tag{16}$$

The final term is zero at an interior optimum and nonnegative at a constrained one. The first term is error temperature cannot remove; the remaining terms measure excess loss from the chosen temperature. This exponential-family identity (Mattei & Loureiro, 2026) also applies when p contains zeros. It does not say what a controller can predict from its available features.

At an interior oracle, $H(q^*) = H(p) + \mathrm{KL}(p\|q^*)$. Matching the teacher's entropy is therefore generally not the KL optimum.

E.4 WHEN QUANTIZATION IS AN EXACT TEMPERATURE CHANGE

If $z = au + b\mathbf{1}$ with $a > 0$, then

$$\mathrm{softmax}\left(\frac{z}{\tau}\right) = \mathrm{softmax}\left(\frac{u}{\tau/a}\right), \qquad T_{\text{correction}} = a\tau. \tag{17}$$

For $a > 1$, native quantization at τ matches a cooler BF16 distribution; for $0 < a < 1$, it matches a warmer one. The correction is exact only in this scaling case, within the allowed range and with the same filtering and tie rule. This exact relation is stronger than matching mean entropy.

Entropy matching and generated responses. Different probabilities and rankings can have the same entropy. The generation-level comparison in Appendix D.12 also includes differences in the histories each model visits.

For the unfiltered temperature family with fixed target p, $\mathrm{KL}(p\|q_\beta) = H(q_\beta) - H(p) + \beta L'(\beta)$. Lower candidate entropy than target entropy therefore favors local warming when feasible; higher entropy alone does not determine the direction. This statement does not extend across changing top-p supports. Lower divergence on sampled prefixes does not guarantee higher task accuracy, shorter responses, or fewer loops: sampled tokens change future prefixes, and BF16 itself can be wrong. Appendix E.8 gives the remaining ranking limits.

E.5 Forward-KL oracle for temperature and top-p

Assume finite quantized logits $z_1 \geq \cdots \geq z_V$ with a fixed tie order; a fixed target p on the same vocabulary; a compact interval $0 < \beta_{\min} \leq \beta \leq \beta_{\max} < \infty$; and unrestricted nucleus cutoff $\rho \in (0, 1]$. Nucleus selection retains the smallest rank prefix whose cumulative unfiltered probability is at least ρ.

Let $S_k = \{1, \ldots, k\}$ and $k^* = \max\{i : p_i > 0\}$. All unfiltered probabilities are positive, so every S_k is attainable at each fixed β. Write $C_k(\beta) = \sum_{i \leq k} \mathrm{softmax}(\beta z)_i$, with $C_0 = 0$. The cutoffs for S_k are precisely $\rho \in (C_{k-1}(\beta), C_k(\beta)]$. Within this nucleus,

$$q_{\beta,k}(i) = \frac{\exp(\beta z_i)}{\sum_{j \leq k} \exp(\beta z_j)} \mathbf{1}\{i \leq k\}.$$

If $k < k^*$, at least one positive target coordinate has zero student probability, giving infinite forward KL. If $k \geq k^*$, the loss is

$$L_k(\beta) = \log \sum_{j \leq k} e^{\beta z_j} - \beta \sum_i p_i z_i - H(p).$$

Adding tokens beyond k^* increases the normalizer and therefore the loss. The joint oracle reduces to minimizing L_{k^*} over the temperature interval, then choosing a cutoff that retains exactly this support. The derivatives are

$$L'_{k^*}(\beta) = \mathbb{E}_{q_{\beta,k^*}}[z] - \mathbb{E}_p[z], \qquad L''_{k^*}(\beta) = \mathrm{Var}_{q_{\beta,k^*}}(z) \geq 0.$$

The temperature is unique when retained logits are nonconstant. If they are equal, every temperature gives the same uniform distribution. Even a unique temperature can have several equivalent cutoffs.

Assumptions and limits. For a full-support teacher, $k^* = V$, so hard truncation cannot improve finite forward KL over the unfiltered student. A restricted cutoff range or finite-precision arithmetic can change which supports are attainable, so this proof does not certify a native sampler. It also does not solve the JSD or reverse-KL problem. The separate JSD numerical search appears in Appendix E.7.

E.6 Forward-KL oracle results

Table 5 uses the cached native BF16 post-filter target at $(0.6, 0.95)$ on all 768 development prefixes. The student calculation uses FP64, descending quantized logits, and ascending token IDs for exact ties. Both scalar and joint temperatures are bounded by $[0.05, 50]$. The scalar comparator leaves student $p = 1$. The same equal task/question/policy/trial/position weights apply to every row. No positive minimum probability or smoothing is imposed on the teacher.

Table 39 retains uncertainty and tie sensitivity. Intervals use 10,000 paired whole-question bootstrap draws, stratified by task, and are exploratory because development reuse and multiple comparisons are not adjusted. Recovery divides weighted aggregate losses, not per-prefix percentages. The raw hard-filtered student has infinite KL on 97 of 768 rows and cannot provide a finite recovery denominator.

Adding top-p removes 45.9% of the temperature-only oracle's residual: 0.0501 nats ($[0.0374, 0.0642]$), or another 15.65 percentage points of original KL ($[14.38, 17.23]$). Four prefixes have order-sensitive boundary ties. The teacher has singleton support at 580 positions and the

Student policy	KL (nats) [95% CI]	Reduction (%) [95% CI]
Unfiltered baseline	0.3201 $[0.2371, 0.4133]$	0.0
Temperature oracle	0.1091 $[0.0851, 0.1351]$	65.9 $[61.4, 69.6]$
Joint oracle	0.0590 $[0.0459, 0.0727]$	81.6 $[77.7, 84.7]$
Joint, whole boundary tie	0.0599 $[0.0465, 0.0739]$	81.3

Table 39: **Forward-KL oracle uncertainty.** All rows use the same post-filter BF16 target. Reductions use the finite unfiltered baseline. The last row retains the complete boundary tie group; its recovery interval is not reported. These mathematical optima are not native-runtime guarantees.

joint oracle at 567; temperature is nonidentifiable on those 567 oracle rows. Returned endpoints therefore do not establish warming or cooling frequencies.

Independent optimization checks on 48 prespecified rows agree within 1.5×10^{-14} nats. Log-domain evaluation avoids probability underflow, but the calculation does not certify a native-sampler optimum.

E.7 JSD ORACLE RESULTS

Table 40 compares oracle action families with the earlier fixed-reference JSD controller from Appendix A.6, not the shared-temperature fits.

Student action family	JSD (nats)
Native settings, $T = .6$, $p = .95$	0.0430
Earlier fixed-reference predictor	0.0407
Per-prefix temperature oracle, $p = .95$	0.0242
Per-prefix joint oracle	0.0151

Table 40: **Development JSD oracle recovery.** Every row uses the same post-filter BF16 target and FP64 mathematical student. Oracles use privileged BF16 information and are numerical optima within the stated tolerance, not certified native-runtime optima. Appendix E.7 gives numerical and support qualifications.

Table 40 uses the same 768 reused development prefixes, cached hard BF16 target, temperature bounds $[0.05, 50]$, and equal question/task weights. All action families use mathematical FP64 student arithmetic, descending-logit supports, and ascending token-ID ties. The temperature oracle here fixes $p = .95$; it differs from the $p = 1$ forward-KL comparator. Intervals use 10,000 paired whole-question bootstrap draws, stratified by task and retaining all prefixes and both fixed seeds. They are unadjusted for development reuse or selection.

Numerical search. JSD need not be convex in inverse temperature. We search $\beta \in [0.02, 20]$ by bisecting the interval with the lowest loss bound. For fixed supports, one lower bound uses tangent envelopes of the concave $\log q_i(\beta)$; another uses a midpoint Taylor expansion with bounded curvature. We take the larger bound and zero, then subtract 2×10^{-12} for floating-point padding.

For a fixed cutoff, endpoint support sizes bound the possible supports; small ranges are enumerated and larger ranges use probability bounds. Joint search considers supports ending at BF16-positive tokens, since removing a trailing BF16-zero block lowers JSD. Whole-boundary-tie supports are checked separately. Search stops at a 10^{-8} gap between the feasible loss and lower bound, or reports a precision failure or the 20,000-split limit. All 6,144 recorded curves met this tolerance. Near support discontinuities, the result may approximate an unattained infimum. These padded FP64 bounds are not formal interval certificates or native-runtime guarantees.

Retaining whole boundary tie groups gives joint JSD 0.01532 instead of 0.01506. The joint oracle excludes BF16-positive probability on 27 rows, with mean excluded mass 0.00355. This is permitted by JSD and would make forward KL infinite on those rows. Temperature is nonidentifiable on 584 joint-oracle rows, including flat supports; representative $T = 50$ values do not demonstrate warming.

The predictor's mathematical JSD is 0.04068, versus 0.04072 under the native sampler. Two seed-specific actions on one MATH prefix account for most of the difference; equal support sizes do not establish identical retained tokens. Its numerical cause remains unresolved.

E.8 Limits of correcting token rankings

Positive temperature preserves logit rankings, and top-p retains a leading set without reordering it. A lower-ranked token therefore cannot become the unique winner through these controls. Ties follow the specified ordering; exact tie-broken greedy decoding is a separate action from the zero-temperature softmax limit, which can distribute mass across ties.

Let a_B and a_Q be the BF16 and quantized greedy tokens. For target δ_{a_B}, forward KL is $-\log q(a_B)$. An exactly greedy candidate δ_{a_Q} gives zero KL when the tokens agree and infinite KL otherwise. If a_B is the quantized winner, singleton filtering gives it probability one at positive temperature. If it has rank r, any nucleus containing it also retains the $r-1$ preceding tokens, each with at least its probability; hence $q(a_B) \le 1/r$. Excluding it gives zero probability. Uniform unfiltered sampling gives log loss $\log V$. Zero and infinite temperature are outside the bounded heads; reaching a finite bound does not establish an unrestricted endpoint optimum.

For $L_t = -\log q_t(a_{B,t})$, mean BF16-token probability is $\mathrm{mean}(e^{-L_t})$, which need not rise when average log loss falls. Neither quantity measures task accuracy. Figure 1 uses a sampled reference distribution rather than a greedy target.

F Use of AI tools

Generative AI assisted with planning, literature review, code and tests, derivations, writing, and figures. LLM coding agents were also used extensively to run and monitor experiments, collect model outputs, and process and analyze the resulting data.

We also tried Sakana AI's ShinkaEvolve[2] to automate the search for improvements to earlier temperature predictors. The evaluated proposals did not provide a clear enough improvement to be adopted in the final method.

The authors remain responsible for the experiments, claims, and manuscript. Automated checks do not replace human verification.

[2] https://github.com/SakanaAI/ShinkaEvolve