QUANTIZING DIFFUSION LLMS UNDER PARALLEL DECODING: COMMITMENT RISK AND THE PARALLELISM TAX

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion large language models (dLLMs) are served with accelerated decoders that cache context and commit, in one step, every masked position whose confidence clears a threshold. Yet post-training quantization of dLLMs has so far been calibrated and evaluated under fixed-step samplers without caching. We study W4A4 quantization under the deployed decoder and find that the decoder determines both where quantization harms accuracy and what it costs. Irreversible errors enter chiefly through forced commitments, made when no position clears the threshold: they are a minority of commitments but carry most quantization-induced flips. Quantization also levies a *parallelism tax*, extra forward passes per response: by compressing top logit gaps, it pushes positions below the threshold while leaving predictions largely intact. We propose DECAL, which reweights layer-wise calibration by a commitment-risk prior measured on the deployed decoder, and an argmax-preserving commit temperature that recovers the tax. On LLaDA and Dream, DECAL attains the best average accuracy among W4A4 methods, with lower run-to-run variance than prior calibration.

1 INTRODUCTION

Diffusion large language models (dLLMs) generate text by iteratively unmasking a response with bidirectional context (Austin et al., 2021a; Sahoo et al., 2024), and open models such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025) now rival autoregressive (AR) models of similar scale. Their speed in practice comes from accelerated decoders: Fast-dLLM (Wu et al., 2026) combines block-wise approximate KV caching with confidence-thresholded parallel decoding, committing in one step every masked position whose confidence exceeds a threshold. Serving cost thus factors into the number of forward passes and the cost of each. Caching and parallel decoding reduce the former; low-bit post-training quantization (PTQ) (Frantar et al., 2023), for instance with rotations (QuaRot; Ashkboos et al., 2024) or learned affine transformations (FlatQuant; Sun et al., 2025), reduces the latter. Ideally, the two savings would compound.

Existing dLLM quantization methods (Lin et al., 2025; Xu et al., 2026; Zhang et al., 2026; Yan et al., 2026; Huang et al., 2026), however, are calibrated and evaluated apart from this decoder, under the model's native sampler, which commits a fixed number of positions per step without caching. The closest, FAIR-Calib (Huang et al., 2026), reweights calibration by a position prior from a deliberately policy-agnostic probe; in our reproduction under Fast-dLLM, this prior is nearly uniform on LLaDA and does not follow where the decoder forces commitments (Figure 1a). Yet under threshold decoding, the model's confidence, passed through the decoding rule, determines both how many forward passes a response takes and which decisions become irreversible.

We study W4A4 quantization directly under Fast-dLLM with dual caching, comparing each quantized model with a full-precision shadow evaluated on identical decoding states. Under the native sampler, the number of steps is fixed in advance and quantization can change only what is written; under threshold decoding, it also changes when positions are written and how many steps a response takes. This yields two observations with no counterpart under fixed-step samplers.

Observation 1: irreversible errors enter through forced commitments. A token is written either because its confidence exceeds the threshold (*threshold commitment*) or because no position in the

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.

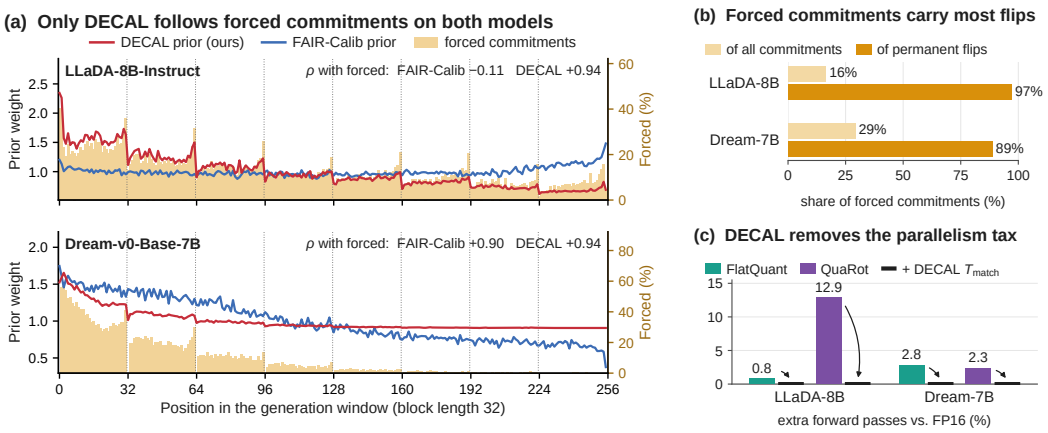


Figure 1: **(a)** Calibration priors of FAIR-Calib and DECAL over the 256 window positions (lines; mean 1), against the share of FP16 trajectories in which each position is a forced commitment under the deployed decoding rule (bars; GSM8K training prompts, no cache; Dream: returned output only); ρ: Spearman correlation. FAIR-Calib's policy-agnostic prior is uncorrelated with forced commitments on LLaDA and, on Dream, tracks their overall decline but not their jumps at block boundaries. **(b)** Forced commitments are a minority of commitments but carry most permanent flips (GSM8K, W4A4 FlatQuant). **(c)** Parallelism tax, the extra forward passes per response relative to FP16 (GSM8K), at $T = 1$ (filled) and with the commit temperature of DECAL at T_{match} (black line), counted as zero at or below FP16. Panels (b, c) use Fast-dLLM with dual cache and $\tau = 0.9$.

active block qualifies and the decoder must still write the most confident one (*forced commitment*); either way, it is never revised. Forced commitments make up only 16–29% of commitments yet carry 89–97% of the *permanent flips*, commitments that quantization changes and that survive into the returned output (Figure 1b, Section 3.2).

Observation 2: quantization levies a parallelism tax. Quantization deflates confidence while largely preserving what the model would write, so fewer positions clear the threshold at each step and each response takes more forward passes. We call this extra cost the *parallelism tax*; it reaches 13% on LLaDA and, as we show, can be recovered without retraining (Figure 1c, Section 3.3).

We therefore align both calibration and decoding with the deployed decoder. DECAL (**DE**coder-aligned **C**ommitment-risk c**AL**ibration) weights layer-wise calibration by a commitment-risk prior estimated from full-precision trajectories of the deployed decoder. A *commit temperature* restores the confidence scale at the threshold test; since it preserves the argmax, it changes when positions are committed, never which token a position receives in a given state.

Our contributions are as follows:

- To our knowledge, we are the first to systematically analyze PTQ for dLLMs under the cached, threshold-based parallel decoder used in deployment, identifying forced commitments, including a placement failure absent from AR decoding (writing the wrong position), as the entry point of irreversible errors, and characterizing the parallelism tax that quantization levies.
- We propose DECAL, which scores each generation position by its flip probability, through the written token or the chosen position, times its reach into the returned output. Collected without any cache, the prior transfers to cached deployment, and its definition applies to both instruction-tuned and base models. DECAL attains the best average accuracy among W4A4 methods on both models, with and without cache, and, under cached decoding, a lower run-to-run variance than FAIR-Calib on every task.
- We introduce an argmax-preserving commit temperature, set by matching full-precision forward passes, that recovers the full tax of QuaRot on LLaDA without a detectable accuracy change and offers a controllable accuracy–efficiency trade-off beyond it.

2 Related Work

Diffusion language models. Masked diffusion models, typically bidirectional Transformers (Vaswani et al., 2017), generate text by iteratively denoising a fully masked sequence (Austin et al., 2021a; Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024). LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025) scale them to billions of parameters. Because vanilla sampling recomputes full bidirectional attention at every step, a line of work accelerates dLLM inference with approximate KV caching (Ma et al., 2025; Liu et al., 2026) and adaptive parallel decoding (Israel et al., 2025); Fast-dLLM (Wu et al., 2026) combines caching with threshold decoding and is the decoder we study.

Post-training quantization of LLMs. Reconstruction-based PTQ fits quantization parameters layer by layer or block by block on a small calibration set (Nagel et al., 2020; Li et al., 2021). For LLMs, weight-only methods use second-order updates (Frantar et al., 2023) or protect salient channels using activation statistics (Lin et al., 2024), while weight–activation methods must handle activation outliers (Dettmers et al., 2022), by smoothing (Xiao et al., 2023), rotation (Ashkboos et al., 2024; Liu et al., 2025), or learned transformations and clipping (Shao et al., 2024; Sun et al., 2025).

Quantization of diffusion models and dLLMs. For image diffusion, calibration must cover the activations of all timesteps (Shang et al., 2023; Li et al., 2023). For dLLMs, Lin et al. (2025) show that AR PTQ methods transfer but W4A4 remains hard because of activation outliers. DLLMQuant (Xu et al., 2026) and Quant-dLLM (Zhang et al., 2026) calibrate across timesteps and mask ratios, STaR-Quant (Yan et al., 2026) treats masked and unmasked token states separately, and FAIR-Calib (Huang et al., 2026) weights calibration positions by a prior probed from a full-precision teacher that commits one random position per step, with neither blocks nor a threshold. None is calibrated or evaluated under the cached threshold decoder used in deployment.

Temperature and confidence. Temperature scaling calibrates predicted probabilities post hoc (Guo et al., 2017; Minderer et al., 2021); we use it to restore the scale the commit threshold reads.

3 Observations

We fix the decoder and the measurement, then establish the two observations our method builds on.

3.1 Decoder and Measurement

Threshold parallel decoding. Confidence-ordered parallel commitment dates back to Mask-Predict (Ghazvininejad et al., 2019) and MaskGIT (Chang et al., 2022); dLLM decoders apply it block by block, left to right (Arriola et al., 2025). A window of K masked positions follows the prompt. At forward pass t, let $\mathcal{A}_t$ be the masked positions of the active block and, for $i \in \mathcal{A}_t$, let $z_t(i)$ be the logits, $p_t(i) = \max_v \mathrm{softmax}(z_t(i))_v$ the confidence, and $m_t(i)$ the gap between the two largest logits. Fast-dLLM (Wu et al., 2026) commits

$$\mathcal{C}_t = \begin{cases} \{\, i \in \mathcal{A}_t : \; p_t(i) \geq \tau \,\} & \text{if non-empty (threshold commitments),} \\ \{c_t\}, \;\; c_t = \arg\max_{i \in \mathcal{A}_t} p_t(i) & \text{otherwise (a forced commitment),} \end{cases} \tag{1}$$

and writes $\hat{x}_t(i) = \arg\max_v z_t(i)_v$ at every $i \in \mathcal{C}_t$; committed tokens are never revised. With dual caching, keys and values outside the active block are cached, refreshed at the first pass of each block (*build step*) and read by the others (*reuse steps*), a distinction that exists only for cached block decoding.

Measurement. We let the quantized model decode and run its full-precision counterpart as a shadow on the same states, without letting it commit. A commitment is *flipped* if the shadow would write a different token or, at a forced commitment, choose a different position; the flip is *permanent* if the token is part of the returned output. We measure W4A4 LLaDA-8B-Instruct and Dream-v0-Base-7B on GSM8K with Fast-dLLM (dual cache, $\tau = 0.9$).

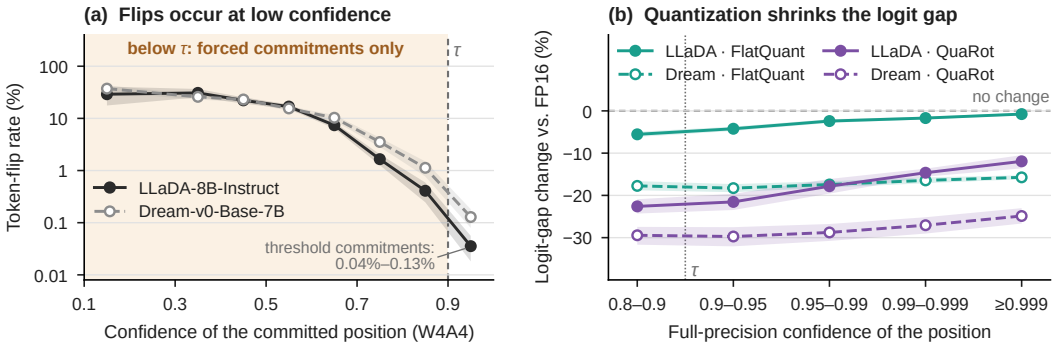


Figure 2: Evidence for the two observations (GSM8K, W4A4, against a full-precision shadow on the same states). **(a)** Token-flip rate, the fraction of commitments whose token differs from the shadow's, by the quantized model's confidence at commitment (FlatQuant). **(b)** Change of the gap between the two largest logits relative to full precision, grouped by full-precision confidence. Bands: 95% confidence intervals.

3.2 Observation 1: Irreversible Errors Enter Through Forced Commitments

The threshold filters decisions by margin. A threshold commitment is made only when the top token leads its runner-up by a sizable logit margin, which a quantization perturbation can overturn only if it exceeds that margin: the threshold admits only robust decisions. The forced rule suspends this filter exactly when the model is uncertain, writing the most confident position at whatever margin it has; since nothing is revised, forced commitments are the channel through which fragile decisions become permanent. Accordingly, the flip rate falls steeply with the confidence at commitment and is up to three orders of magnitude higher below τ, a range only forced commitments reach (Figure 2a). With FlatQuant, forced commitments are 16.5% of all commitments on LLaDA and 29.2% on Dream, yet carry 97.1% and 88.8% of the permanent flips (Figure 1b): per commitment, 6.0% and 9.8% of them flip, against 0.04% and 0.13% of threshold commitments.

How forced commitments fail, and why only here. A forced commitment can flip its token, as in AR decoding, but also its position: when the most confident positions are nearly tied, quantization can change which one is written, a placement flip that AR decoding, with its fixed order, cannot make. Either error is permanent and becomes context for every later pass, so its damage grows with how long it is read before the returned output is complete. By contrast, a fixed-step sampler commits its k most confident positions whatever their margins, spreading fragile commitments over the whole schedule, and a random-commit probe, as in FAIR-Calib, does not see them at all (Figure 1a). Under threshold decoding, they are concentrated, can be located on full-precision trajectories before quantization, and differ in how far their errors reach; the prior of Section 4.1 measures both.

3.3 Observation 2: Quantization Levies a Parallelism Tax

An absolute test on a relative quantity. The threshold reads confidence as an absolute number, but confidence is set by the gaps between the top logit and its competitors: with two candidates, $p \geq \tau$ requires a lead of 2.2 nats at $\tau = 0.9$ (Section 4.2). Anything that shrinks these gaps without changing the top token delays commitments while leaving every decision intact, and quantization does exactly this. It adds nearly independent noise to the logits, which barely changes their spread but pulls the top logit toward the rest and lifts the strongest of the many competitors, so the gaps near the peak shrink most: on LLaDA with QuaRot, the top-2 gap shrinks by 26% while the standard deviation of the logits is unchanged (Section B), and across models and quantizers the shrinkage reaches 30% at confident positions (Figure 2b).

From compressed gaps to extra passes. Positions near τ then fall below it: fewer positions clear the threshold at each step, more steps end in a forced commitment, and each response needs more forward passes, up to 13% more on LLaDA with QuaRot (Figure 1c; both tasks in Section A).

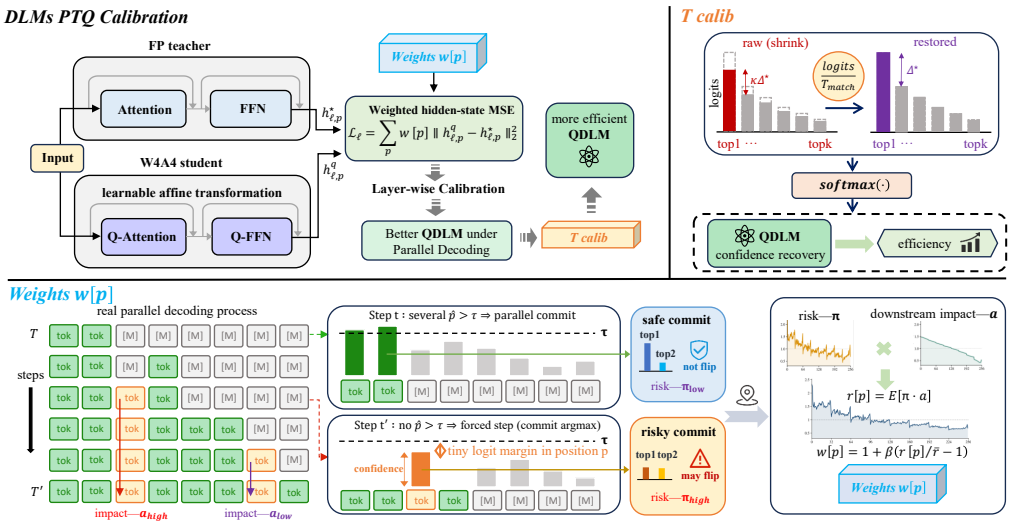


Figure 3: Overview of DECAL. **Top left:** a W4A4 student is calibrated layer by layer against the full-precision teacher with a hidden-state MSE weighted by $w[p]$, and decodes with the commit temperature T_{match}. **Bottom:** how $w[p]$ is obtained. Under the deployed parallel decoder, positions committed above τ are safe (low flip risk π), whereas forced commitments at small logit margins may flip (high π); positions committed early are read by more later passes (high reach a); $w[p]$ follows from $r[p] = \mathbb{E}[\pi \cdot a]$. **Top right:** dividing the logits by T_{match} restores the top-logit gaps that W4A4 compresses, and with them the full-precision commit rate.

We call this cost the *parallelism tax*. It compounds Observation 1, since the deflation that slows decoding also routes more decisions through the forced channel.

Deflation, not lost capability. The shadow of Section 3.1 shows that the tax is a matter of scale. The width loss it measures step by step matches an independent end-to-end measurement (13.4% against 14.6% on 400 questions; Section A), so the extra passes come from fewer commitments per step, not from diverging trajectories; and where the shadow would commit but the quantized model withholds, the two predict the same token in 97.9% of cases, against 68.3% over all masked positions. Quantization perturbs predictions broadly but, where decisions are made, only lowers confidence, so rescaling confidence recovers the tax without touching written tokens (Section 4.2).

4 METHOD

4.1 DECAL: COMMITMENT-RISK CALIBRATION

Trajectories. DECAL decides *where* the calibration budget is spent (Figure 3, bottom and top left). The full-precision model decodes J unlabeled prompts under rule (1) with the deployed threshold, block length, and window, without cache. Position p of trajectory j is committed once, at pass $t_j(p)$, with logit gap $m_j(p)$; L_j is the returned length after the stop rule (EOS or task stop sequence).

Flip probability. A commitment fails if the runner-up token overtakes the written one (*token flip*) or, at a forced commitment, if the runner-up position overtakes the chosen one (*placement flip*). Both are log-probability differences: the logit gap m and the placement gap $g = \log p_t(c_t) - \max_{i \neq c_t} \log p_t(i)$. Modeling each difference's quantization perturbation as zero-mean Gaussian with scale s gives flip probabilities $\Phi(-m/s)$ and $\Phi(-g/s)$; for independent failures,

$$\pi_j(p) = 1 - \Big(1 - \Phi\big(-\tfrac{m_j(p)}{\tilde{m}}\big)\Big)\Big(1 - \Phi\big(-\tfrac{g_j(p)}{\tilde{m}}\big)\Big), \tag{2}$$

with $g_j(p) = +\infty$ except at forced commitments. With the quantizer not yet fixed, we set $s = \tilde{m}$, the median forced-commitment gap in returned outputs; only the ranking of π matters, as (4) normalizes it.

Reach. A flipped token damages the returned output directly if returned, and indirectly through every later pass that reads it while returned tokens are still being written, up to $t_j^\star = \max_{p'<L_j} t_j(p')$:

$$a_j(p) = \sqrt{\mathbf{1}[p < L_j] + D_j(p)}, \qquad D_j(p) = \big[\, t_j^\star - t_j(p) \,\big]_+. \tag{3}$$

The square root is the root-mean-square accumulation of approximately independent perturbations, a premise we verify in Section F. Reach gives no budget to a base model's discarded continuation, so one definition serves base and instruction-tuned models.

Prior and weights. Averaging over trajectories,

$$r(p) = \frac{1}{J}\sum_{j=1}^{J} \pi_j(p)\, a_j(p), \qquad w(p) = 1 + \beta\Big(\frac{r(p)}{\bar{r}} - 1\Big), \qquad \beta = \frac{c}{\mathrm{CV}(r)}. \tag{4}$$

As π is the probability of an error and a the damage it does, their product, not either factor alone, is an expected loss: if flipping and reaching are independent, $r(p)$ is the expected damage to the returned output, and it vanishes if either factor does. The affine map preserves the ranking and sets $\mathrm{CV}(w) = c$, a concentration budget trading focus against the effective sample size $K/(1+c^2)$.

Weighted layer-wise calibration. As in FAIR-Calib (Huang et al., 2026), the prior reweights FlatQuant's layer-wise reconstruction (Sun et al., 2025) on generic text of length T_c whose last K positions stand in for the window:

$$\min_{\Theta_\ell} \sum_{n=1}^{N}\sum_{i=0}^{T_\mathrm{c}-1} \omega_i \left\| h_\ell(\mathbf{x}_{n,i}) - \hat{h}_\ell(\mathbf{x}_{n,i}; \Theta_\ell) \right\|_2^2, \tag{5}$$

with $\omega_i = \lambda\, w(i - T_\mathrm{c} + K)$ on the last K positions (window scale λ) and a context weight ϵ elsewhere; $h_\ell, \hat{h}_\ell$ are full-precision and quantized layer outputs, Θ_ℓ the learnable transformations and clipping thresholds. As the cache approximates attention, not the decoding rule, the prior transfers to cached decoding: collecting it with dual cache instead, from a slightly smaller prompt set, leaves GSM8K accuracy on LLaDA unchanged (76.45 ± 0.24 against 76.65 ± 0.46, three runs each).

4.2 Commit Temperature

The threshold is a margin. The temperature sets *when* positions commit (Figure 3, top right). With top token $v^\star$ and gaps $\Delta_v = z_{v^\star} - z_v \geq 0$, the confidence and two-candidate test are

$$p = \Big[1 + \sum\nolimits_{v \neq v^\star} e^{-\Delta_v}\Big]^{-1}, \qquad p \geq \tau \iff \Delta \geq m_\tau := \log \tfrac{\tau}{1-\tau}. \tag{6}$$

The threshold thus fixes a margin in nats ($m_\tau = 2.20$ at $\tau = 0.9$) on the full-precision logit scale. If quantization compresses the top gaps as $\Delta^\mathrm{q} \approx \kappa\Delta$ with $\kappa < 1$ (Figure 2b), a position commits only when $\Delta \geq m_\tau/\kappa$: the effective threshold rises, which is the parallelism tax.

Definition. Because quantization compresses the logits whatever the decoder, the temperature applies to any confidence-thresholded decoder: rule (1) uses $p_t^T(i) = \max_v \mathrm{softmax}\big(z_t(i)/T\big)_v$ for the threshold test and for c_t, while the written token stays $\arg\max_v z_t(i)_v$. We study it in the deployed setting, cached block decoding, where we apply it on reuse steps only. Build steps keep $T = 1$: a position withheld there disagrees with full precision more often than at reuse steps (3.2% against 1.8%), and a premature commitment there becomes context for the entire block (Section E).

Proposition 1. *For any decoding state and any $T \in (0, 1]$ applied as above (proof in Section F):*

(i) content invariance: *every committed position receives the same token as at $T = 1$;*
(ii) monotone admission: *the number of committed positions is non-increasing in T, and a position admitted at T but not at $T = 1$ satisfies $Tm_\tau \leq \Delta < m_\tau$ (two candidates);*
(iii) exact restoration: *if the logits contract uniformly, $z^\mathrm{q} = \kappa z + b\mathbf{1}$, then $p^T(z^\mathrm{q}) = p(z)$ at $T = \kappa$.*

Calibration by matching the commit rate. Contraction is not uniform across positions and confidence levels (Figure 2b), so no single κ can be read off the logits. A threshold decoder, however, consumes confidence only through τ-crossings, whose aggregate is the number of forward passes; we calibrate T on exactly this statistic, the one for which τ was set:

$$T_{\mathrm{match}} = \sup\left\{\, T \in (0,1] :\ \bar{N}_{\mathrm{q}}(T) \le \bar{N}_{\mathrm{fp}}(1) \,\right\}, \tag{7}$$

where $\bar{N}_M(T)$ is the mean number of forward passes per response of model M on calibration prompts. The criterion is *label-free* (it never consults accuracy), *consistent* (it returns κ under uniform contraction, Proposition 1(iii)), *self-disabling* (a quantizer without tax receives $T_{\mathrm{match}} = 1$), and *cheap*: by Proposition 1(ii), $\bar{N}_{\mathrm{q}}$ decreases with T, so two evaluations bracketing the target give T_{match} by linear interpolation. One temperature is calibrated per model–quantizer pair and reused on every task. Like temperature scaling (Guo et al., 2017), (7) fits one scalar, but to the statistic its consumer uses: a classifier's temperature is fitted to the likelihood of labels, whereas a threshold decoder only asks whether confidence crosses τ, so we fit the rate at which it does. With two candidates, a temperature commits when $\Delta \ge Tm_\tau$, as a lower threshold would, and at matched forward passes the two are indistinguishable (Section E): (7) adds not a new knob but a label-free way to set it.

5 Experiments

5.1 Setup

Models, decoding, and tasks. We quantize LLaDA-8B-Instruct (Nie et al., 2025) and Dream-v0-Base-7B (Ye et al., 2025) and decode with Fast-dLLM (Wu et al., 2026) using dual cache, $\tau = 0.9$, block length 32, and 256 generated tokens; we also report threshold decoding without cache. Following Fast-dLLM, we evaluate GSM8K (5-shot, flexible-extract exact match; Cobbe et al., 2021), HumanEval (0-shot, pass@1; Chen et al., 2021), MBPP (3-shot, pass@1; Austin et al., 2021b), and MATH (Hendrycks et al., 2021) with the 4-shot Minerva prompt (Lewkowycz et al., 2022), 200 problems per subject, and an answer-equivalence check. These match Fast-dLLM's official evaluation; we report accuracy rather than perplexity, which scores teacher-forced text and never runs the decoder.

Methods. All methods quantize weights and activations to 4 bits (W4A4). QuaRot (Ashkboos et al., 2024) represents rotation-based PTQ, and FlatQuant (Sun et al., 2025) uses its default calibration with uniform weights. FAIR-Calib (Huang et al., 2026) and DECAL share FlatQuant's reconstruction on 128 WikiText-2 sequences (Merity et al., 2017) of length $T_{\mathrm{c}} = 1{,}024$ whose last $K = 256$ positions form the window, and differ only in the weights ω of (5): the probed prior for FAIR-Calib (reproduced in our pipeline) and the commitment-risk prior (4) for DECAL, collected without cache from 5-shot GSM8K training prompts (about 640 per model; FAIR-Calib keeps its default 512, beyond which its own ablation shows no gain). This one-time cost is 2.8 GPU-hours on Dream (153 forward passes per prompt; LLaDA needs about half as many), fewer passes in total than FAIR-Calib's 512 probes of 256 steps each. FAIR-Calib, a prior over positions, is the dLLM calibration closest to ours; DLLMQuant and Quant-dLLM (2-bit, weight-only) instead change the calibration data or the quantizer, and are orthogonal to the prior. DECAL uses $(\epsilon, c) = (0.01, 0.30)$ with FAIR-Calib's window scale on LLaDA, and $(1, 0.15)$ with unit-mean window weights on Dream. The commit temperature uses GSM8K forward-pass counts only (full runs for LLaDA, 50 prompts for Dream). Under dual cache, we repeat calibration three times for the FlatQuant-based methods and report the mean and standard deviation; FP16 and QuaRot are single runs. Decoding without cache is substantially slower, and we run each configuration once. All accuracy tables use $T = 1$; we evaluate the commit temperature in the deployed setting, with dual cache (Section 5.4).

5.2 Main Results

Table 1 reports the deployed setting. Rotation alone is not enough at W4A4: QuaRot loses 3.9 average points on LLaDA and 18.9 on the base model Dream, whereas the FlatQuant-based methods retain most of the full-precision accuracy. Among them, DECAL has the best average on both models and the best GSM8K, HumanEval, and MATH accuracy; on MBPP, the three lie within one standard deviation of one another. Although the prior is collected on GSM8K prompts alone, its gains carry

Table 1: W4A4 accuracy (%) under the deployed decoder (Fast-dLLM, dual cache, $\tau = 0.9$); FlatQuant-based methods: mean $\pm$ std over three calibration runs; best W4A4 result in bold.

Model	Method	GSM8K	HumanEval	MBPP	MATH	Avg.
LLaDA-8B Instruct	FP16	78.70	36.59	36.40	30.79	45.62
	QuaRot	74.30	33.54	32.20	26.86	41.73
	FlatQuant	75.44 $\pm$1.07	35.57 $\pm$3.97	35.80 $\pm$2.43	28.21 $\pm$0.22	43.76
	FAIR-Calib	75.84 $\pm$0.88	35.77 $\pm$1.86	**36.12** $\pm$1.41	28.43 $\pm$2.29	44.04
	DECAL	**76.65** $\pm$0.46	**36.21** $\pm$0.93	36.08 $\pm$1.06	**28.67** $\pm$0.24	**44.40**
Dream-v0 Base-7B	FP16	73.16	53.05	51.00	30.79	52.00
	QuaRot	53.30	23.78	37.80	17.64	33.13
	FlatQuant	66.34 $\pm$1.05	40.85 $\pm$3.66	**47.40** $\pm$1.39	23.57 $\pm$0.97	44.54
	FAIR-Calib	66.47 $\pm$1.47	40.88 $\pm$4.25	46.33 $\pm$2.19	23.43 $\pm$0.86	44.28
	DECAL	**67.27** $\pm$1.08	**41.81** $\pm$2.11	46.20 $\pm$1.78	**24.60** $\pm$0.72	**44.97**

Table 2: W4A4 accuracy (%) with threshold decoding without cache ($\tau = 0.9$; one run per configuration).

	LLaDA-8B-Instruct					Dream-v0-Base-7B				
Method	GSM8K	HumanEval	MBPP	MATH	Avg.	GSM8K	HumanEval	MBPP	MATH	Avg.
FP16	78.09	42.07	40.20	32.00	48.09	72.40	49.39	53.80	31.86	51.86
QuaRot	73.46	34.76	38.40	28.21	43.71	54.89	29.88	39.20	18.86	35.71
FlatQuant	76.35	33.54	**40.60**	29.64	45.03	66.41	35.37	47.20	24.50	43.37
FAIR-Calib	76.27	38.90	40.21	29.57	46.24	64.06	**43.90**	46.40	25.57	44.98
DECAL	**76.95**	**39.02**	39.60	**29.80**	**46.34**	**68.16**	43.29	**52.00**	**26.64**	**47.52**

over to code, as it encodes where the decoder commits fragile tokens, a property of the decoding schedule rather than of the domain. DECAL is also the most stable calibration: its standard deviation across runs is below FAIR-Calib's on all eight model–task pairs, often by about half (0.46 against 0.88 on LLaDA GSM8K, 2.11 against 4.25 on Dream HumanEval, 0.24 against 2.29 on LLaDA MATH). Without cache (Table 2), the regime in which its prior is collected, DECAL again has the best average on both models, with the widest margin on Dream (47.52 against 44.98 for FAIR-Calib), so its gains do not hinge on the cache.

5.3 ABLATION: COMPONENTS OF THE PRIOR

Table 3 isolates the two factors of (4). Every risk-aware prior improves on uniform weights, by 1.1–1.2 average points on LLaDA and 1.4–1.8 on Dream: where the calibration budget goes matters. How the factors share it depends on the model and the task, as follows from what each measures: π marks *locally fragile* commitments, a commitments whose errors *propagate* through later reads.

Task: local versus propagating errors. GSM8K solutions are short, so correctness hinges on a few decisive tokens. MATH solutions are long derivations in which an early error propagates into everything written after it. On Dream, the factors separate: π alone carries the GSM8K gain (+1.8 points, against none for a alone), whereas a alone gives the largest MATH gain (+2.8 against +1.7).

Table 3: Components of the prior under dual cache (mean of two calibration runs). Uniform sets $w \equiv 1$, single-factor variants set the other factor of (4) to one; c and reconstruction are shared.

	LLaDA-8B-Instruct			Dream-v0-Base-7B		
Prior	GSM8K	MATH	Avg.	GSM8K	MATH	Avg.
Uniform ($w \equiv 1$)	74.91	28.22	51.57	66.00	23.11	44.56
Flip probability only ($a \equiv 1$)	76.04	**29.32**	52.68	**67.78**	24.81	46.30
Reach only ($\pi \equiv 1$)	76.17	29.11	52.64	65.97	**25.89**	45.93
DECAL ($\pi \cdot a$)	**76.65**	28.85	**52.75**	67.74	24.90	**46.32**

Model: why the separation appears on Dream. An instruction-tuned model stops on its own, so its commitments, fragile or not, lie inside the returned answer: on LLaDA the factors largely agree, and either alone recovers most of the gain (+0.9 to +1.3 points). A base model writes past its answer, so part of its fragile commitments falls in a discarded continuation while reach concentrates on the answer: on Dream the factors single out different positions, and the task decides which matters. The product is the robust choice: DECAL need not know which failure mode a task exhibits, has the best average on both models, and stays within one point of the best variant everywhere, whereas each factor alone falls short somewhere.

5.4 CALIBRATING THE COMMIT TEMPERATURE

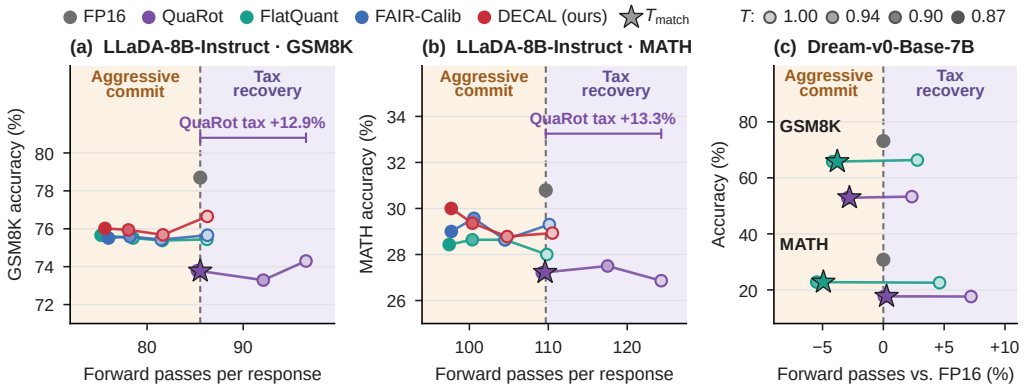


Figure 4: Commit temperature: accuracy against forward passes (**a, b**: LLaDA-8B-Instruct; **c**: Dream-v0-Base-7B, relative to FP16). Colors: methods; darker shades: lower T; stars: T_{match} (7), interpolated (omitted where it is ≈ 0.99, i.e., at $T = 1$); dashed line: FP16 at $T = 1$.

Calibration recovers the tax where it exists, and only there. We evaluate each LLaDA configuration at $T \in \{1, 0.94, 0.90, 0.87\}$ and set T_{match} by (7) from GSM8K forward-pass counts alone, holding out MATH. QuaRot pays a 13% tax on both tasks (Figure 4a, b). At its $T_{\text{match}} = 0.873$, it returns to the full-precision forward-pass count, while accuracy moves by less than 0.6 points, within single-run noise (± 2.4); solving (7) on MATH alone gives 0.874, so the GSM8K-calibrated temperature carries over to the held-out task. FlatQuant, FAIR-Calib, and DECAL pay under 1%, receive $T_{\text{match}} \approx 0.99$, and are left essentially unchanged.

Below T_{match}, it is an explicit trade-off. At $T = 0.87$, these three models need about 11% fewer forward passes than FP16, with accuracy within about one point of $T = 1$ across the ladder.

It transfers across models. On Dream, T_{match} calibrated on only 50 GSM8K prompts removes taxes of 2–7% on both tasks with accuracy changes below 0.6 points (Figure 4c); the small calibration set slightly overestimates the tax, leaving up to 5% fewer forward passes than FP16. Lowering T only reorders commitments: at each question's first early release, the written token is the one the $T = 1$ schedule eventually writes in at least 98.7% of questions (Section C).

6 CONCLUSION

We studied W4A4 quantization of dLLMs under the decoder they are served with. Under cached threshold decoding, quantization errors become irreversible chiefly through forced commitments, and quantization levies a parallelism tax by deflating confidence rather than changing predictions. DECAL weights calibration by commitment risk, and a commit temperature calibrated on forward passes alone recovers the tax without retraining or labels. On LLaDA and Dream, DECAL attains the best average accuracy among W4A4 methods with lower run-to-run variance, and the temperature restores full-precision decoding cost. The prior is tied to the deployed block length and threshold, and we have studied one decoder family at one bit width; extending commitment-aware calibration to other decoders and lower bit widths is a natural next step.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing, including drafting, editing, and polishing the text, and with parts of the code, including analysis and plotting scripts. We have reviewed all AI-assisted work: the authors checked and revised all text, and verified and tested all AI-assisted code; every reported number comes from our own experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 5.1 lists the models, decoder settings, tasks, calibration data, and hyperparameters of every method; Sections 4.1 and 4.2 define the prior, the weighted reconstruction, and the calibration of the commit temperature; Sections A to E describe the measurement protocols behind the observations and the temperature study; and Section F proves Proposition 1. We will release the code for collecting the prior, the weighted calibration, the commit temperature, and the evaluation.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

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.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

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. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021.

Haoyu Huang, Linlin Yang, Sheng Xu, Boyu Liu, Guodong Guo, Zhongqian Fu, Hang Zhou, and Baochang Zhang. FAIR-Calib: Frontier-aware instability-reweighted calibration for post-training quantization of diffusion large language models. In *International Conference on Machine Learning*, 2026.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion LLMs via adaptive parallel decoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17535–17545, 2023.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021.

Haokun Lin, Haobo Xu, Yichen Wu, Ziyu Guo, Renrui Zhang, Zhichao Lu, Ying Wei, Qingfu Zhang, and Zhenan Sun. Quantization meets dLLMs: A systematic study of post-training quantization for diffusion LLMs. *arXiv preprint arXiv:2508.14896*, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyan Wei, Shaobo Wang, Yichen Zhu, and Linfeng Zhang. dLLM-Cache: Accelerating diffusion large language models with adaptive caching. In *International Conference on Machine Learning*, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dKV-Cache: The cache for diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? Adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, 2020.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In *International Conference on Machine Learning*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In *International Conference on Learning Representations*, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023.

Chen Xu, Zhixuan Chen, and Dawei Yang. DLLMQuant: A post-training quantization framework tailored for diffusion-based large language models. In *International Conference on Machine Learning*, 2026.

Xin Yan, Aqiang Wang, Zhenglin Wan, Xingrui Yu, and Ivor Tsang. STaR-Quant: State-time consistent post-training quantization for diffusion large language models. *arXiv preprint arXiv:2606.04945*, 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Tianao Zhang, Zhiteng Li, Xianglong Yan, Haotong Qin, Yong Guo, and Yulun Zhang. QuantdLLM: Post-training extreme low-bit quantization for diffusion large language models. In *International Conference on Learning Representations*, 2026.

A On-Policy Paired Measurement of the Parallelism Tax

Comparing the forward passes of two free-running decoders mixes two effects: quantization lets fewer positions clear τ at each step, and the two decoders visit different states after their first disagreement. Only the first is caused by quantization error. After reporting the end-to-end tax of every configuration, this appendix isolates the first effect on LLaDA-8B-Instruct with QuaRot, the configuration used throughout Appendices A–D.

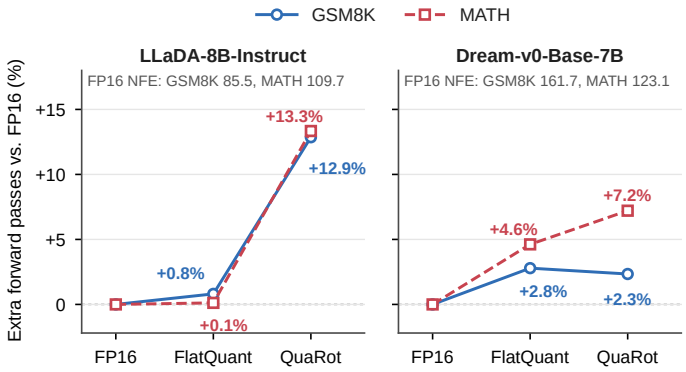


Figure A.1: End-to-end parallelism tax on both tasks: extra forward passes per response relative to FP16 at $T = 1$ (Fast-dLLM, dual cache, $\tau = 0.9$). Its magnitude depends on the model, the quantizer, and the task.

End-to-end tax. Figure A.1 extends Figure 1c to MATH. The tax appears on every model, quantizer, and task, with a magnitude that ranges from 0.1% (FlatQuant on LLaDA, MATH) to 13.3% (QuaRot on LLaDA, MATH).

Setup. The quantized model has exclusive control of decoding and produces states $x_1, \ldots, x_S$. At every step, a full-precision shadow runs one read-only forward pass on bit-identical input; for QuaRot, it is a copy of the model taken after the rotations are folded in and before the weights are quantized. It keeps its own KV cache and never commits, so any difference between the two arms at a state is due to numerical precision alone.

Estimator. The width of model m at state x is the number of tokens it would write,

$$w^m(x) = \max\big(\,|\{i \in \mathcal{M}(x) : p_i^m(x) \geq \tau\}|,\ 1\big),$$

where $\mathcal{M}(x)$ is the set of masked positions of the active block and $\max(\cdot, 1)$ encodes the forced commitment. Width is a count rather than a set because the forced rule fixes how many positions are written, not which: each arm forces its own most confident position, and a mismatch in position costs no parallelism. Averaging over the trajectory, $W^m = \frac{1}{S}\sum_t w^m(x_t)$, the paired tax is $\hat{\mathcal{T}} = W^{\mathrm{fp}}/W^{\mathrm{q}} - 1$: generating L tokens takes $S = L/W$ steps, so $\hat{\mathcal{T}}$ is the relative excess of steps at the quantized width. It uses only same-state comparisons, and the identity $\sum_t w^{\mathrm{q}}(x_t) = L$ checks the recording exactly.

Table A.1: The paired estimate and the end-to-end tax, measured on disjoint data.

	Paired on-policy	Independent end-to-end
Data	32 prompts, 2,910 steps, 47,919 paired positions	400 questions, two complete runs
Reading	$W^{\mathrm{fp}} = 3.193$, $W^{\mathrm{q}} = 2.815$	$\mathrm{NFE}^{\mathrm{fp}} = 85.23$, $\mathrm{NFE}^{\mathrm{q}} = 97.64$
Tax	$\hat{\mathcal{T}} = 13.4\%$	$\mathrm{NFE}^{\mathrm{q}}/\mathrm{NFE}^{\mathrm{fp}} - 1 = 14.6\%$

The tax is per-step width loss. The two estimates in Table A.1 share no data and differ by 1.2 points. Since $\hat{\mathcal{T}}$ consists of per-step width loss only, the forward-pass gap is explained by fewer commitments per step rather than by trajectory divergence. The small residual is expected: $\hat{\mathcal{T}}$ is conditioned on the states the quantized policy visits, whereas the full-precision decoder follows its own trajectory, on which positions deferred at one step can be released together later.

Withheld positions keep their content. Let R be the positions the shadow would commit but the quantized model withholds ($N = 1{,}700$). The two models predict the same token on 97.94% of R (96.84% at build steps, 98.16% at reuse steps), against 68.31% over all masked positions: quantization perturbs predictions broadly, but mostly at low confidence, where no commitment is decided. Because R is selected by "full precision confident, quantized not", we also condition on the quantized side only, using the decision band $p^{\mathrm{q}} \in [\tau - 0.15, \tau + 0.05]$; agreement is 97.52%. The withheld parallelism thus consists of commitments whose content is not in dispute, which is what makes it recoverable. Whether recovering it is accuracy-neutral is a separate question, answered against the quantized model at $T = 1$ (Figure 4).

B Where Quantization Compresses the Logits

Probe. At the first build step of block 0, all generation positions are masked and both models receive bit-identical inputs, so their logits can be paired position by position (GSM8K, 400 questions; HumanEval, 164). Two independent estimators quantify the compression. The geometric one sorts each position's logits and fits a single through-origin scale $d_j^{\mathrm{Q}} = \kappa_{\mathrm{gap}}\, d_j^{\mathrm{FP}}$ to the gaps $d_j = L_{(1)} - L_{(j)}$, $j = 2, \ldots, 64$; it sees only shape. The behavioural one solves exactly, over the full vocabulary, for the temperature θ with $\max \mathrm{softmax}(L^{\mathrm{FP}}/\theta) = p_1^{\mathrm{Q}}$; it sees only the outcome, and $1/\theta$ is the equivalent compression.

Table B.1: Compression by full-precision confidence (medians per stratum). QR: QuaRot; NR: W4A4 without rotation (group size 128); G: GSM8K; H: HumanEval.

		positions			κ_{gap}			$1/\theta$
p_1^{FP}	$p_1^{\mathrm{FP}} \to p_1^{\mathrm{Q}}$ (QR-G)	QR-G	NR-G	QR-H	QR-G	NR-G	QR-H	QR-G
$[0.0, 0.3)$	$0.115 \to 0.110$	9,306	9,299	4,511	0.974	0.924	1.002	0.989
$[0.3, 0.6)$	$0.419 \to 0.342$	2,533	2,533	648	0.908	0.883	0.906	0.881
$[0.6, 0.8)$	$0.678 \to 0.559$	740	748	86	0.905	0.890	0.885	0.857
$[0.8, 0.9)$	$0.847 \to 0.729$	153	151	0	0.895	0.899	–	0.842
$[0.9, 1.0]$	$0.936 \to 0.872$	68	69	0	0.899	0.895	–	0.835

Table B.2: Compression by scale (GSM8K). Rank-wise rows (top-2 to top-64) pool positions with $p_1^{\mathrm{FP}} \geq 0.3$ ($n = 3{,}494$); the remaining rows use all positions ($n = 12{,}800$). κ_{all} is the OLS slope of $L_u^{\mathrm{Q}} = \alpha + \kappa_{\mathrm{all}} L_u^{\mathrm{FP}} + \nu_u$ over 1,024 token ids fixed by a seed and independent of both models; $\sigma_\nu / s_{\mathrm{FP}} = \sqrt{(s_{\mathrm{Q}}/s_{\mathrm{FP}})^2 - \kappa_{\mathrm{all}}^2}$ follows from $s_{\mathrm{Q}}^2 = \kappa_{\mathrm{all}}^2 s_{\mathrm{FP}}^2 + \sigma_\nu^2$.

Scale	Quantity	QuaRot	NR (g128)
top-2	$d_2^{\mathrm{Q}}/d_2^{\mathrm{FP}}$	0.740	0.743
top-3	$d_3^{\mathrm{Q}}/d_3^{\mathrm{FP}}$	0.802	0.803
top-5	$d_5^{\mathrm{Q}}/d_5^{\mathrm{FP}}$	0.843	0.838
top-10	$d_{10}^{\mathrm{Q}}/d_{10}^{\mathrm{FP}}$	0.878	0.863
top-64	κ_{gap}	0.906	0.885
1,024 fixed tokens	κ_{all}	0.965	0.942
full vocabulary	$s_{\mathrm{Q}}/s_{\mathrm{FP}}$	1.007	0.992
full vocabulary	$\sigma_\nu/s_{\mathrm{FP}}$	0.288	0.311

Compression is systematic. Confidence drops in every stratum of Table B.1, and in the decision-relevant strata ($p_1^{\mathrm{FP}} \geq 0.3$) it has the same size for both quantizers and both tasks, so it is a property

of the model and quantizer rather than of one rotation scheme or prompt distribution. Its consequence shows on the probe itself: of the positions full precision would commit, only 23 of 68 (QuaRot) and 46 of 69 (no rotation) still clear τ after quantization, a small-sample illustration of direction rather than a measure of the tax.

Compression is concentrated at the peak. Table B.2 shows compression weakening monotonically as the scale widens: the gap between the peak and its strongest competitor shrinks to 0.74, while the slope over model-independent tokens is close to one and the standard deviation of the whole logit vector is unchanged. Quantization adds nearly independent noise that offsets the shrinkage in the bulk of the vocabulary, but thins exactly the few gaps on which the maximum softmax probability depends.

Implication for the temperature. Dividing the logits by T scales every gap by $1/T$, so a temperature can undo this compression. Because the compression depends on scale and on the confidence stratum, no single number in these tables is the right temperature, and we calibrate it on the commit rate instead (Section 4.2).

C Temperature Ladder: Mechanism and Premature Commitment

Table C.1: Instrumented temperature ladder (GSM8K test set; temperature on reuse steps). NFE and single-sequence throughput are relative to $T = 1$ of the same run. F: forced steps per question; $\bar{k}$: mean width of threshold steps. Early release: share of generated positions committed earlier than at $T = 1$. First-release agreement: share of questions whose token released at the first early release equals the one the $T = 1$ schedule eventually writes. Slot agreement: share of output slots identical to $T = 1$ (trailing EOS padding excluded).

T	NFE	F	$\bar{k}$	throughput	early release	first-release agr.	slot agr.
1	–	45.81	4.02	–	–	–	–
0.94	-5.9%	40.74	4.18	$+6.0\%$	5.0%	99.4%	70.4%
0.87	-12.6%	35.06	4.36	$+12.9\%$	11.2%	99.0%	52.1%
0.84	-15.3%	32.69	4.44	$+14.1\%$	14.0%	98.7%	45.7%

Recovery reverses the cause. With K generated positions, F forced steps, and threshold steps of mean width $\bar{k}$, the number of forward passes is $\mathrm{NFE} = K/\bar{k} + F\,(1 - 1/\bar{k})$. The tax raises F and lowers $\bar{k}$; the temperature does the opposite. From $T = 1$ to $T = 0.84$, forced steps fall by 28.6% and the threshold width rises by 10.4%, i.e., the merges that quantization suppressed happen again. Forward passes and measured throughput move together at every temperature (e.g., -12.6% and $+12.9\%$ at $T = 0.87$), so in the single-sequence setting NFE is a faithful proxy for latency.

Premature commitment is rare and mostly one step early. Since Proposition 1 fixes what is written at a given state, the only risk the temperature introduces is committing a position before its context is complete. We measure it along the $T = 1$ trajectory: at every visited state and for every position that $T = 0.87$ would release there, we compare the token released at that state with the token the $T = 1$ schedule eventually writes. Both are recorded values, so the comparison involves no approximation; it characterizes the intervention itself rather than its downstream course, since the two schedules separate after the first early release. Four in five such releases are advanced by a single step, and these agree in 99.9% of cases. Agreement decreases monotonically with the advance, to about one half, but advances of that depth account for less than 0.1% of releases. The first-release agreement in Table C.1 is restricted to the step at which both schedules provably occupy the same state and is therefore exact and on-policy.

Why outputs still change. Slot agreement falls from 70.4% to 45.7% without contradicting Proposition 1: the proposition concerns what is written at a given state, whereas slot agreement compares the endpoints of two different trajectories. Once a position is committed earlier, its neighbours face a different context at the same step, so outputs diverge while every write remains the argmax of its own state. Proposition 1 rules out that the temperature makes the model write wrongly, not that it changes the output.

D Compression and Suppression Along the Trajectory

Table D.1: On-policy paired recording (32 GSM8K prompts, 2,910 steps, 47,919 paired positions), stratified by the filled fraction ρ of the active block ($\rho = 0$ at build steps). FP / Q commit: positions above τ in each arm. Suppression: share of full-precision commitments the quantized arm withholds. Share of pool: share of all 1,700 withheld positions. $\kappa^\star$: median of the per-position equivalent compression, solved exactly.

Stratum	paired positions	FP commit	Q commit	suppression	share of pool	$\kappa^\star$
build ($\rho = 0$)	8,192	2,362	2,157	12.1%	16.8%	0.930
reuse, $\rho \in [0, .25)$	15,293	1,571	1,274	24.8%	22.9%	0.966
reuse, $\rho \in [.25, .50)$	12,906	1,607	1,324	24.1%	22.8%	0.930
reuse, $\rho \in [.50, .75)$	7,991	1,548	1,231	26.5%	24.1%	0.921
reuse, $\rho \in [.75, 1)$	3,537	1,033	873	22.0%	13.4%	0.912
all	47,919	8,121	6,859	20.9%	100%	0.939

The block-0 probe is representative. Section B reads the compression at the first build step of block 0, the only point at which both arms see bit-identical inputs, whereas the temperature acts on reuse steps. Along a whole trajectory, the median $\kappa^\star$ spans only 0.912–0.966 across strata (Table D.1), and the build-step value (0.930) lies in the same neighbourhood as the trajectory median (0.939) and the probe's decision-relevant values (0.895–0.908).

Reading the table. The suppression ratios are position-level upper bounds, since at a forced step each arm commits its own most confident position without any loss of parallelism; the tax itself is measured by the counting estimator of Section A.

E Design Choices of the Commit Temperature

Build steps: least to gain, most to lose. Table E.1 combines the paired recording of Sections A and D. A response has one build step per block, 8 of roughly 90 forward passes, and these steps hold only 285 of the 1,700 withheld commitments (16.8%), because build steps are the least suppressed stratum. Releasing a withheld commitment writes a token that full precision would not write with probability one minus the agreement rate, 3.2% at build steps against 1.8% at reuse steps; weighted by the number of releases, build steps would contribute 25.7% of the expected miswrites. Each of them, moreover, is written into a fully masked block and becomes context for the other 31 positions in every later step, whereas a release at a late reuse step affects only the few positions still masked. Build steps therefore keep $T = 1$.

Table E.1: Expected benefit and cost of releasing the withheld commitments, by step type (paired recording of Section A). Expected miswrites: withheld positions $\times$ (1 $-$ agreement).

Step type	withheld positions	share of gain	agreement	expected miswrites	share of miswrites
build	285	16.8%	96.84%	9.0	25.7%
reuse	1,415	83.2%	98.16%	26.0	74.3%

Relation to lowering τ. With two candidates, $p^T \geq \tau$ holds exactly when $\Delta \geq T m_\tau$, so on the original confidence a global temperature acts like the threshold $\tau_{\text{eff}} = \sigma(T m_\tau)$, which is 0.858 at $T = 0.82$ and $\tau = 0.9$. To first order, the temperature is therefore a reparametrization of τ, and Table E.2 confirms that the two cannot be told apart at matched forward passes. Interpolating the temperature ladder to the forward-pass count of $\tau = 0.82$ predicts its forced steps and threshold width to within 0.1% (30.84 against 30.82, and 4.566 against 4.564), and both settings reach the same accuracy. We therefore do not claim a new decoding knob. What (7) contributes is how to set this knob without labels: the value it returns reads as the quantizer's compression ($T = \kappa$ under uniform contraction, Proposition 1), is calibrated from forward-pass counts alone and carries over across tasks (Section 5.4), and leaves τ at the value its deployer chose for the full-precision model.

Table E.2: Lowering the temperature and lowering the threshold (LLaDA-8B-Instruct with QuaRot, GSM8K, 400 questions; same instrumentation as Table A.1; the temperature acts on reuse steps, the threshold on every step). F: forced steps per question; $\bar{k}$: mean width of threshold steps.

Setting	NFE	F	$\bar{k}$	accuracy (%)
FP16, $\tau = 0.9$	85.23	39.05	4.698	78.25
W4A4, $\tau = 0.9$, $T = 1$	97.64	46.16	4.076	76.50
W4A4, $\tau = 0.9$, $T = 0.87$	85.04	35.13	4.426	76.00
W4A4, $\tau = 0.9$, $T = 0.82$	80.88	31.48	4.545	75.75
W4A4, $\tau = 0.82$, $T = 1$	80.16	30.82	4.564	75.75

Global versus stratum-wise temperatures. For every position there is an exact restoring temperature $\kappa^\star$ (Table D.1), but computing it requires the full-precision logits, which are unavailable once the model is deployed in quantized form. The realizable alternative assigns each stratum of Table D.1 its own temperature, its median $\kappa^\star$. We compare the two on the paired recording at matched budgets of premature commitments, i.e., positions the corrected quantized arm would commit but full precision would not, and count the withheld commitments each recovers. At three budgets, the stratum-wise temperatures recover 0.99, 1.01, and 1.00 times as many as the global one. The stratum carries information about the compression, but not about which positions lie just below the threshold, which is all a threshold decoder acts on.

F Proof of Proposition 1 and the Premise of the Reach

Proof of Proposition 1. (i) The argmax is invariant to positive scaling, so dividing the logits by $T > 0$ leaves every written token unchanged. (ii) Replacing Δ_v by Δ_v/T in (6) shows that p^T grows as T decreases, so the threshold set only grows, and a non-empty one commits at least as many positions as a forced commitment; with two candidates, $p < \tau \leq p^T$ is equivalent to $Tm_\tau \leq \Delta < m_\tau$. (iii) The softmax is shift-invariant, so $z^{\mathrm{q}} = \kappa z + b\mathbf{1}$ gives $\mathrm{softmax}(z^{\mathrm{q}}/\kappa) = \mathrm{softmax}(z + b\mathbf{1}/\kappa) = \mathrm{softmax}(z)$.

Premise of the reach. The square root in (3) is the root-mean-square accumulation of approximately independent perturbations across the passes that read a committed token. We verified the independence premise: without the shared per-layer bias, cross-position error terms are 3.6% of the diagonal ones. The choice of the concave map matters little: $\log(1+\cdot)$ in place of the square root gives a prior with correlation 0.9975.