

State-of-the-Art 3-Bit Post-Training Quantization

Invariant Research

18 August 2026

A confidential customer request: Qwen3-0.6B at 3-bit weights, 8-bit activations, symmetric grids, weight groups of 256. This note reports the build and the evidence. The short version:

- **State-of-the-art 3-bit.** Our 3-bit build holds 32.4% GSM8K accuracy in think mode and 26.4% in nothink mode. Stock GPTQModel scores 0.4% and 1.2% at 3-bit, and 29.6% and 7.2% at 4-bit. Our 3-bit model beats their 4-bit model.
- **Pure post-training quantization.** No weight training. About 10M calibration tokens. Training-based routes to comparable low-bit models spend 350B to 4T tokens: four to five orders of magnitude more data than us.
- **Lossless at 4-bit.** The identical pipeline at 4-bit matches fp16 within evaluation noise: 58.0% vs. 59.6% think, 36.0% vs. 34.8% nothink.

1 The spec, and why it is hard

The spec is hardware-driven. Take one output row r of a quantized linear layer with per-group scale $s_{r,g}$ and zero-point $z_{r,g}$. The stored weight for column i in group g is the integer $q_{r,i}$; the dequantized weight is $\hat{w}_{r,i} = s_{r,g} (q_{r,i} - z_{r,g})$. The layer computes

$$y_r = \sum_i \hat{w}_{r,i} x_i = \sum_g s_{r,g} \left(\underbrace{\sum_{i \in g} q_{r,i} x_i}_{\text{integer dot product}} - z_{r,g} \underbrace{\sum_{i \in g} x_i}_{\text{activation sum}} \right). \quad (1)$$

A symmetric grid sets $z_{r,g} = 0$: the activation-sum term disappears, and per (row, group) the kernel saves one multiply-accumulate, one reduction, and the zero-point storage. The group size sets how long the integer pipeline runs before a floating-point rescale, so rescale work and scale traffic amortize as $1/g$ per weight:

group size	32	64	128	256
scale metadata (fp16, bits/weight)	0.500	0.250	0.125	0.0625

Group 256 is the cheapest common footprint to serve and the hardest to quantize. Libraries and papers default to groups of 128 (GPTQModel, AWQ, MiniCPM4 releases) or finer (ExLlama presets, llama.cpp k-quants) because a single scale covering 256 channels leaves nothing to absorb per-channel variation. Every result below is measured at the customer’s group-256 footprint. Similarly, small, heavily trained models are the hardest quantization targets. In the GPTQ evaluation, 3-bit quantization costs +95% perplexity at 125M parameters against +4% at 175B (Frantar et al., 2023), and precision scaling laws show post-training quantization degradation grows with pretraining tokens per parameter (Kumar et al., 2024). Qwen3-0.6B is pretrained on 36T tokens (Qwen3 report), roughly 60,000 tokens per parameter: an extreme point on that axis.

2 Data

Calibration uses 4,000 sequences of length 4,000 from our curated mix: long reasoning traces (mathematics, competitive programming, science), web text, code, and function-calling data. That is roughly 10M tokens, consumed in forward passes only. No gradient step ever touches the weights. The training-based alternative is expensive. ModelBest’s BitCPM4 ([arXiv:2506.07900](https://arxiv.org/abs/2506.07900)) reaches low-bit weights by quantization-aware training: 350B tokens of QAT on top of an 8T-token pretrain. BitNet trains its ternary model from scratch on 4T tokens ([Ma et al., 2025](#)).

route to a low-bit model	tokens	vs. ours
Invariant PTQ (this note)	~10M	1×
BitCPM4 QAT phase (ModelBest)	350B	35,000×
BitNet ternary, from scratch	4T	400,000×
BitCPM4 end to end (pretrain + QAT)	8.35T	835,000×

The same gap holds in kind, not just degree: QAT needs a training cluster, optimizer state, and learning-rate schedules. Our pipeline needs a pretrained checkpoint and a few GPU-hours of calibration.

3 Results

We compare against the fp16 teacher (Qwen/Qwen3-0.6B, WikiText-2 perplexity 20.96) and stock GPTQModel 7.1.0, the standard open-source PTQ library, run at its `desc_act=True` setting, symmetric, group 256, on the same calibration set. The protocol is identical for every model and run once: WikiText-2 perplexity at sequence length 2048, KL divergence to the fp16 teacher, and GSM8K (8-shot, $n=250$) in *think* (≤ 1024 tokens) and *nothink* (≤ 400 tokens) modes. All quantized builds use 4-bit or 3-bit weight codes, group size 256, 8-bit embeddings and LM head. Appendix B adds a matched-format control that isolates the contribution of our correction pass. At 4-bit

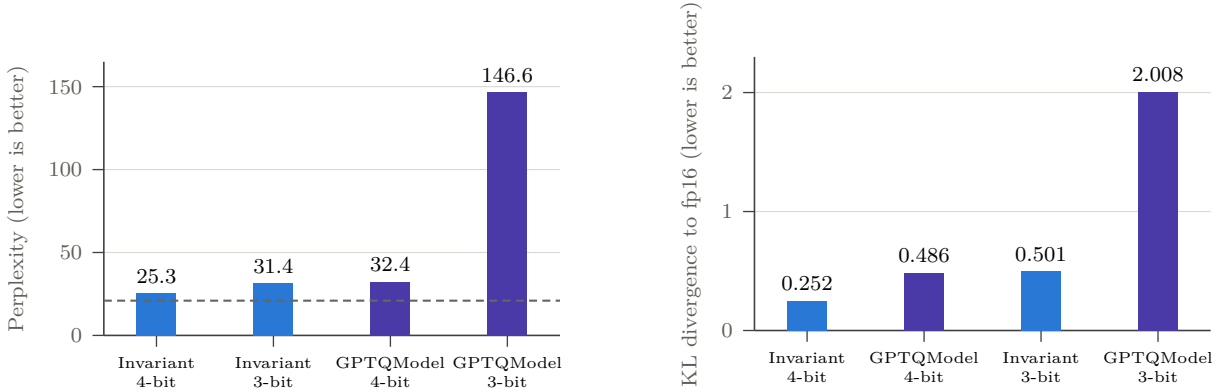


Figure 1: Fidelity at group 256, WikiText-2, sequence length 2048, bars in ascending order. Left: perplexity, dashed line at the fp16 reference (20.96). Right: KL divergence to the fp16 teacher.

our build is lossless within noise and holds half the KL of stock GPTQModel (0.252 vs. 0.486). Stock GPTQModel already loses half the model’s reasoning at 4-bit. At 3-bit it collapses outright: perplexity 146.6, KL 2.008, and over half of its GSM8K generations fail to parse (141/250 think, 119/250 nothink). Our 3-bit build stays intact and beats the stock 4-bit build on perplexity (31.4 vs. 32.4), think accuracy (32.4% vs. 29.6%), and nothink accuracy (26.4% vs. 7.2%), at one fewer bit per weight code. To our knowledge these are the strongest 3-bit results reported for this model, and they hold at the coarsest common group size.

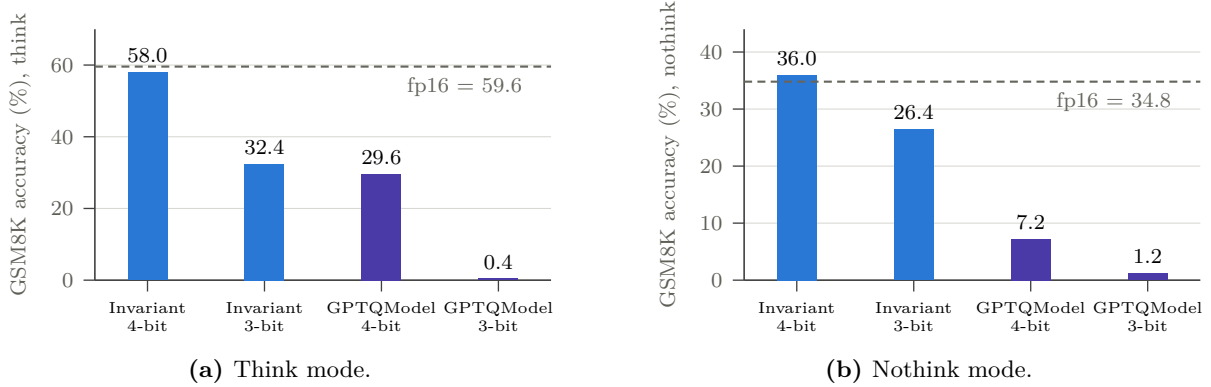


Figure 2: GSM8K accuracy at group 256, bars in descending order. Dashed lines: fp16 reference.

Artefacts, reproduction scripts, and commit SHAs are in the private Invariant repository. Appendix A reproduces the customer’s technical due-diligence questionnaire. Appendix B isolates what our correction pass adds against a matched-format control. Appendix C measures the 8-bit activation path. Appendix D shows the result does not depend on hyperparameter tuning.

A Customer technical due-diligence questionnaire

The customer’s hardware team runs a formal compatibility review before adoption. Their questionnaire is reproduced below, with identifying details redacted. The target configuration is W3A8 with an INT4 KV cache on the customer’s SRAM-CIM edge SoC; each answer is scored pass, conditional pass, or red flag, and any red flag in the first two categories rejects the solution.

1. Numerical format and executability.

- Is the 3-bit quantization standard uniform quantization, or does it rely on codebooks or vector quantization?
- Is the quantization symmetric, or asymmetric with zero-points?
- Are Hadamard transforms or rotation preprocessing used? Offline or online?
- What is the group size? Can it align with 256? How large is the scale metadata?
- Can a pure 3-bit bitplane format be provided in which 256 weights equal 96 bytes, and can it pass a bit-exact golden test?

2. Activations and KV cache.

- For the reported results, are activations running in FP16 or INT8?
- Is KV-cache quantization supported? The production setup uses KV INT4.
- Is the vendor willing to retest under the customer’s evaluation methodology: W3A8, 512 to 2048 samples, perplexity plus downstream tasks, with a loss target of at most 5 to 10%?

3. Model scale and evidence.

- Can actual 3-bit results be provided on Qwen2.5-1.5B and Llama3.2-3B? Smaller models are usually harder to quantize well.
- Are the reported numbers measured on held-out datasets or on the calibration data? Beyond coding benchmarks, how much degradation is there in perplexity and general tasks?
- Which layers remain at high precision? What is the effective average bit-width? Can per-layer precision breakdowns be provided?

4. CIM non-idealities.

- How much accuracy margin remains on real CIM hardware with sense-amplifier offset, drift, and saturation?
- Is the vendor willing to integrate the customer’s noise model into the calibration loop for hardware-in-the-loop co-design?
- Does the system remain stable under SS corners, high temperature, low VDD, and IR drop?

5. Toolchain and IP integration.

- Is the deliverable an open artifact package (weights, scales, precision tables), or is everything tied to a proprietary runtime?
- Are outputs reproducible, diffable, and hashable?
- Who owns the optimized model IP? Can the customer burn it into chips and ship products with it?
- Is this an offline delivery, or does the vendor’s compiler or runtime stay permanently in the loop? Where is the handoff boundary?

6. Real energy efficiency.

- Is the vendor willing to evaluate in fJ per useful MAC operation, rather than compression ratio alone?
- Does the 3-bit model execute in true 3-cycle bit-serial form, or is it compressed for storage and reconstructed to higher precision for computation?

7. Maturity and risk.

- Are there production customers or real silicon, edge, or CIM experience?
- If the vendor disappears, can the customer independently maintain the format?
- How does a roadmap toward 2-bit quantization align with hardware fixed at W3, W4, and W8?
- Is the solution exclusive, or can competitors access the same technology?

B Matched control: Bit-plane GPTQ

Bit-plane GPTQ is the matched control: identical weight format, smoothing, calibration data, and evaluation as our build, with our curvature-guided correction switched off. This isolates what our method adds.

3-bit build	PPL ↓	KL ↓	think % ↑	nothink % ↑
Bit-plane GPTQ (control)	40.12	0.688	21.6	8.8
Invariant	31.37	0.501	32.4	26.4
effect of the correction	−22%	−27%	+10.8 pp	+17.6 pp

Table 1: Matched-format control at 3-bit, group 256. The correction pass is the only difference between the rows.

C 8-bit activations

The spec quantizes activations to int8 (per-token dynamic, symmetric). This is independent of the weight pipeline and cheap: measured on our 3-bit pipeline it costs +0.3 perplexity and +0.01 KL (30.54 to 30.82 PPL, 0.516 to 0.525 KL, group-128 build). The figures of Section 3 isolate the weight comparison.

D Recipe robustness

Figure 3 plots GSM8K accuracy surfaces over the recipe’s hyperparameters: the solver regularization and the correction damping λ . Every point is an independent quantization of the raw model evaluated with the protocol of Section 3. The comparison is between the two quantizer families, both with our correction pass: the uniform-grid quantizer (grey) and the production bit-plane quantizer (blue). For further sweep results, contact the authors.

The gap between the families is a family property, not a tuning artifact. The uniform-grid family never exceeds 20.0% think anywhere on its grid; at the same points the bit-plane family sits at 27 to 36%, and at think even its worst iteration choice clears the uniform-grid sheet everywhere. Without the correction both families stall: 0.0 to 4.4% think for the uniform grid, 19.2% at best for the bit-plane quantizer.

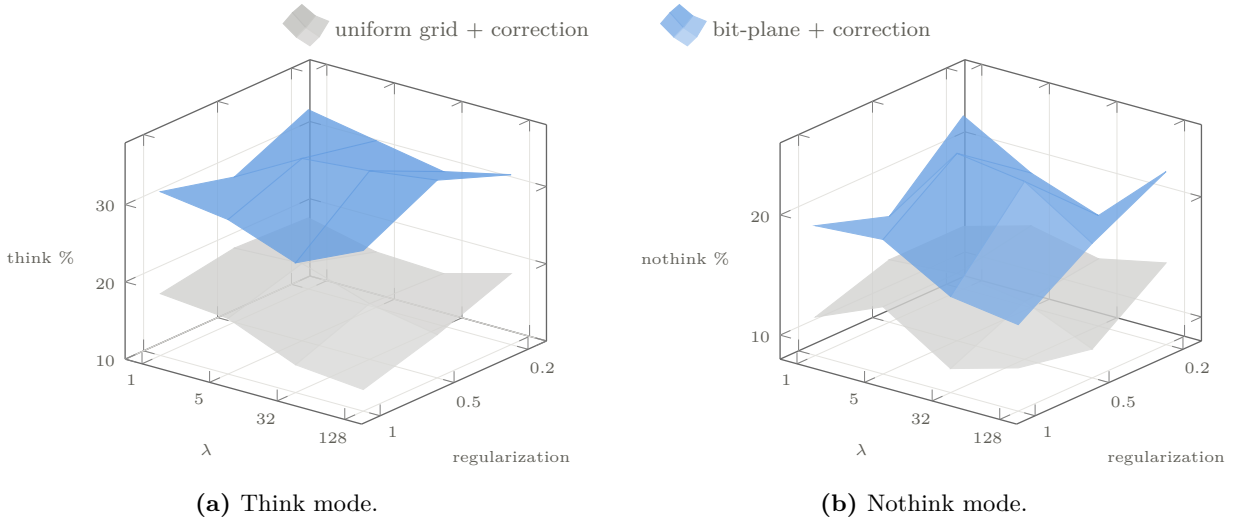


Figure 3: GSM8K accuracy for the two quantizer families, both with the correction pass, on the raw model over the shared regularization and λ grid. Bit-plane shown at its balanced iteration setting.

Findings:

- *The recipe sits on a plateau.* GSM8K noise at $n=250$ is $\sigma \approx 3$ pp per cell. 19 of 36 bit-plane cells are within 2σ of the best think score, and λ is nearly flat (think means 29.6, 29.6, 28.7, 29.6 across $\lambda = 1, 5, 32, 128$). No knob needs fine tuning.
- *KL and PPL do not rank reasoning.* The best-KL cells of the sweep are not the best GSM8K cells. We chose the deployed configuration on GSM8K, balancing think and nothink, not on fidelity metrics.
- *Both ingredients are necessary.* Tuned over their own grids, the uniform-grid quantizer with the correction caps at 20.0% think and the bit-plane quantizer without it caps at 19.2%; composed, the bit-plane family reaches 24 to 36% across its grid.
- *The grid edges were probed.* Extending the regularization and the iteration count one step beyond the swept range in each direction made every reasoning metric flat or worse; all optima are interior to the grid.