Behaviour-Relearned Quantization

Recovering one-bit mixture-of-experts weights by reserving structure and relearning expert codes

Bad Theory Labs technical report

Bad Theory Labs
August 2026

Abstract. BRQ is a research lane for one-bit-class mixture-of-experts compression. The central hypothesis is that below two bits per weight, useful models are not produced by allocating a few more bits to sensitive matrices. The high-precision structure must be reserved by rule, and the low-bit expert codes must be relearned with training. We tested this hypothesis on OLMoE-1B-7B by binarizing only routed expert tensors with group-128 sign and scale, then training the expert latents through a hard binary forward pass using windowed straight-through KL distillation. The unrecovered model was effectively dead: 3.81% teacher top-1 agreement and student perplexity 18,446 against teacher perplexity 9.76 on the fixed Wikitext slice. One windowed pass reached 47.85% top-1, and a two-pass V1a run reached 49.38% with KL 1.448 and student perplexity 39.13. A dense Qwen3-8B control did not show the same regime: all-linear binary recovery stalled near 22%, and an MLP-only spine-reserved control finished at 23.0%. The result supports expert granularity as the cheap recovery regime. It does not support a release claim. Stored free-running generations remained repetitive or incoherent, the V1a receipt did not include a packed binary artifact or SparseGPT/LDLQ initialization, and no run met the pre-registered 80% behavior-promotion gate.

1 Why this paper exists

The previous Bad Theory Labs compression papers answered different questions. *Behavior Before Perplexity* showed that a compressed model can preserve token statistics while losing the behavior that matters to an agentic release. *Range Before Representation* showed that, for BTL-4 Compact, a 35.1B mixture-of-experts model could be released as a 9.96 GB stock-format GGUF when range selection and four expert-weight levels were enough.

BRQ asks the harder question that *Range Before Representation* explicitly did not answer: what happens below the four-level floor? In that paper, ternary experts retained 87.3% under one simulated configuration, but binary experts collapsed to 0.8% or 0.0%. Post-training range selection had reached its floor. Any one-bit result would need an intervention that changes the quantized weights themselves, not a better placement of static levels.

The working thesis became simple:

> At about one bit per weight, bits are not allocated. Structure is reserved, and the surviving code is relearned.

This is not a claim that BRQ has already produced a usable one-bit model. It is a report on the evidence that made BRQ the correct research lane, the experiments that supported it, and the boundaries that prevented promotion.

1.1 Contributions

This report makes four claims, in descending order of confidence:

1. Static post-training quantization is not the mechanism that gets this research line to one bit. The BTL-4 level-count ablation shows a sharp two-level collapse under the tested post-training route.
2. A real MoE with binary routed experts can relearn a large amount of teacher-forced structure from a near-dead floor using hard-forward STE distillation.
3. The cheap recovery regime appears to be expert granularity rather than reserved-spine alone. The Qwen3-8B controls are the key evidence for this correction.
4. The next BRQ run should be judged by behavior and receipts, not by teacher-forced slope. V1a is useful because it failed that boundary clearly.

The paper does not claim novelty over all QAT, KD, mirror descent, binary LLM, or Bonsai-style work. It records a BTL-specific method and evidence trail for one-bit MoE recovery.

2 Background: what BAQ and RBR ruled out

The allocation-first path was BAQ, Behaviour-Allocated Quantization. BAQ tested whether low-bit retention could be bought by placing scarce precision where end-to-end behavior or KL sensitivity said it mattered. It produced a real positive result at moderate rates, then failed as the target moved below two bits.

At moderate rates, allocation was real. On a Qwen2.5-0.5B proxy, an end-to-end KLD selector beat uniform placement at matched rate on paired behavior. That result matters because it validates the broader BTL habit of choosing compression decisions from downstream behavior, not from local weight error.

At the lower edge, the same axis stopped paying. In the BTL-4 work, the dominant variable became the usable range inside a four-level group. Once the representation dropped below four levels, the remaining question was no longer where to place extra bits. There were no extra bits to place.

On the BTL-4 MoE, *Range Before Representation* found that four expert-weight levels were still viable, three levels were damaged but not dead, and two levels

collapsed. The relevant rows are:

Table 1: BTL-4 simulated post-training quantization results that motivated BRQ.

Configuration	Levels	bpw	Retention
2-bit experts g128 + 4-bit remainder	4	2.40	95.8%
2-bit experts g256 + 3-bit remainder	4	2.21	93.2%
ternary experts + 4-bit	3	2.03	87.3%
ternary experts + 3-bit	3	1.96	69.5%
binary experts only	2	2.29	0.8%
binary experts + 4-bit	2	1.47	0.0%

That table is the handoff point. The static post-training route was not merely weak at one bit. It was dead. A one-bit MoE could not be argued from bit allocation alone.

The second clue came from public Bonsai artifacts. The deployed binary representation was not a hidden vector codec. It was ordinary-looking groupwise sign and scale at roughly 1.125 physical bits per matrix weight. Small coordinate probes suggested the binary signs were not simple nearest rounding of the public source weights. The interpretation used here is cautious: the endpoint is scalar, but the quality-producing stage is upstream of packing and training-like. Public evidence did not disclose the full Bonsai transformation, so BRQ does not claim to reproduce it.

2.1 What earlier low-bit failures taught

Several older BTL experiments are relevant because they explain why BRQ does not promote on reconstruction, token agreement, or aggregate slope.

- Scalar hard-ternary recovery reached high teacher-forced token agreement in earlier BTL V2 work while retaining only a minority of sealed free-running BFCL behavior.
- A scalar binary backbone with some transformer weights rescued at INT4 retained abstention but emitted no valid tool calls.
- A packed Bonsai backbone plus an unchanged BTL adapter did not turn into a BTL model. The adapter added size and latency without adding the missing behavior.
- BAQ's small-model allocation win did not become a dense one-bit recipe. The stronger evidence was the negative result: low-bit agent behavior cannot be inferred from token metrics.

Those failures are not background color. They set the promotion rule for BRQ. Any result that recovers top-1 but loops in generation is a research signal, not a release.

3 Method

BRQ separates structure from recovery.

3.1 Reserve the spine

In an MoE, most parameters live in routed experts, while a small fraction controls embeddings, attention, routing, normalization, shared components, and output interfaces. For Kimi K3, the measured ledger contained 2,779,931,837,184 parameters. Routed experts accounted for 2,722.7B of them, or 97.94%. Holding the non-routed spine at source MXFP4 while storing routed experts in a binary group-128 container projects to 413.9 GB, or 1.191 all-in bits per weight:

Table 2: K3 projection used to motivate the BRQ architecture.

Component	Params	Rate	Bytes
spine	57.19B	MXFP4 4.25	30.4 GB
routed experts	2,722.7B	binary 1.125	382.9 GB
container	–	measured overhead	0.66 GB
total	2.78T	1.191 bpw	413.9 GB

This is a structural rule, not a greedy allocation result. The spine is small enough to keep because it defines routing and model interface behavior. The expert tensors are the byte problem.

The reserve rule has two jobs. The first is byte accounting: if the spine is only about two percent of K3 parameters, keeping it high precision barely moves the all-in rate. The second is functional stability: routing and interface tensors decide which knowledge path the model enters, how tokens are embedded, and how the final distribution is read. Corrupting those structures can change the problem rather than merely adding noise to an expert computation.

The reserve rule is not the full mechanism. The Qwen control in Section 5 shows that a reserved dense spine does not make a dense binary model recover cheaply. Reservation keeps the MoE problem well-posed. Expert granularity makes the recovery tractable.

3.2 Binarize only the routed experts

The BRQ probe used groupwise binary sign and scale on the contraction axis of expert tensors. For a group size of 128, the expert payload rate is:

$$1 + \frac{16}{128} = 1.125 \text{ bits per weight.}$$

During training, latent floating-point expert weights are maintained, but the forward pass uses the hard quantized weights through a straight-through estimator. The export target remains the binary code and scale container. No LoRA, residual matrix, or dense fallback is part of the claimed deployment form.

This distinction is central. During research, the latent weights are allowed to move because the optimizer needs a surface. During deployment, the model must be represented by signs and scales. A BRQ run that keeps an fp16 adapter, a dense residual, or a shadow weight in the inference graph is no longer a one-bit expert result.

3.3 Windowed recovery

The implementation trains overlapping layer windows front to back. In each window, only the selected expert latents require gradients. The rest of the model remains in its quantized-forward state, so each window sees the accumulated error from the already-recovered prefix and unrecovered suffix.

The loss is full-vocabulary KL from a frozen teacher to the student over the sampled sequence positions. V1a added a 50/50 mix of Wikitext and Glaive tool-calling text plus an EOS multiplier, but the audit later found that the multiplier did not receipt the realized loss mass. Future BRQ runs must log the actual token counts and weighted loss share for EOS, tool names, argument keys, delimiters, and error-recovery spans.

3.4 The intended BRQ ladder

The method was designed as a ladder, not as a single jump to K3:

Table 3: BRQ validation ladder.

Rung	Purpose	Status
V0	Prove binary expert recovery has a large slope on a real MoE.	Done
V1a	Add more passes and task-like calibration; audit receipts.	Done, not promoted
V1b	Matched A/B on token budget vs corrected objective.	Required
V2	Full artifact accounting on a small MoE.	Open
V3	Larger MoE scale point before 35B/K3 spend.	Deferred
K3	Mission-scale one-bit-class MoE recovery.	Gated

The ladder exists to stop the story from driving the experiment. If V1b cannot fix free-running behavior at small scale, the correct outcome is to redesign the objective or stop, not to buy a larger run.

4 Experimental setup

The main testbed was `allenai/OLMoE-1B-7B-0924`, a real routed MoE that fit on a single A100-80GB. The probe quantized routed expert tensors only. The teacher remained bf16. The student loaded in fp32 latents and used hard binary expert forwards during both evaluation and training.

The V0 configuration used sequence length 512, batch 4, 256 calibration chunks, 6 evaluation chunks, 48 generation tokens, layer window 3, stride 2, 75 steps per window, learning rate 10^{-4}, and seed 20260730. V1a used the same model and target, two passes, a tool-calibration mix, and `eos_weight=4.0`. Both runs evaluated on the same six fixed Wikitext chunks, 3,072 next-token positions.

The control model was `Qwen/Qwen3-8B`. Two control arms were recorded. The first binarized all main linears and OOM-crashed after window 16–21, leaving a partial receipt. The second binarized only MLP linears while reserving attention and the rest of the spine. It completed one pass. These controls are not architecture-matched to OLMoE, so they are used to test the interpretation, not to rank models.

4.1 Implementation details

The implementation patch replaces each target module's forward pass with a hard-quantized weight view:

$$q_g = \alpha_g \cdot \mathrm{sign}(u_g), \quad \alpha_g = \mathrm{mean}(|u_g|).$$

The straight-through estimator returns gradients to the latent u_g while preserving hard binary weights in the forward pass. The student therefore trains the codes it will eventually export, rather than training a dense proxy and hoping the final projection behaves.

The evaluation computes full-vocabulary KL, top-1 agreement with teacher logits, student perplexity, and teacher perplexity over a small fixed Wikitext evaluation slice. The generation probes are deliberately simple and deterministic. They are not a broad benchmark. Their job is to catch obvious degeneration before a numerical curve is mistaken for behavior.

4.2 Receipt quality

The V0 and V1a JSON receipts store configuration, ledger, per-checkpoint evals, generation samples, losses, and training time. They do not store enough for a release claim. The V1a audit identified missing model revisions, missing code hashes, missing realized critical-token mass, and no packed artifact hash. This paper treats those omissions as part of the result because they decide what can be said publicly.

5 Results

5.1 OLMoE V0: dead to partially recovered

The binary expert floor was near-dead. Top-1 agreement with the teacher was 3.81%, KL was 7.395, and student perplexity was 18,446 compared with teacher perplexity 9.76. After one front-to-back pass, top-1 agreement reached 47.85%, KL fell to 1.456, and student perplexity fell to 39.13.

Table 4: OLMoE V0 recovery trajectory.

Checkpoint	Top-1 vs teacher	KL	Student ppl
floor	3.81%	7.395	18,446
after layers 0–2	36.20%	2.431	104.53
after layers 4–6	45.57%	1.641	47.61
after layers 10–12	47.43%	1.475	40.24
after layers 14–15	47.85%	1.456	39.13

This is the result that made BRQ worth preserving. A binary expert MoE did not stay dead under recovery training. The slope was large, monotone enough to matter, and achieved with a small run rather than a full-model backward pass.

The run took 2,061.7 seconds of training time. It used 1,200 window steps in total: 16 layers, layer windows of 3, stride 2, and 75 steps per window. With batch 4 and 512-token windows, the rough training exposure was about 1.23M token positions. The evaluation denominator was much smaller, 3,072 positions, so the numerical precision should not be confused with a broad language-model benchmark.

5.2 OLMoE V1a: more steps, same warning

V1a doubled the pass count and added a tool-calibration mix plus a nominal EOS weighting. It reached 49.38% top-1, KL 1.448, and student perplexity 39.13. The second pass gained only 3.26 percentage points over the end of pass one.

Table 5: OLMoE V1a two-pass result.

Checkpoint	Top-1 vs teacher	KL	Student ppl
floor	3.81%	7.395	18,446
end pass 0	46.13%	1.626	47.10
mid pass 1	48.40%	1.514	41.88
end pass 1	49.38%	1.448	39.13

The V1a audit is the reason this paper does not claim promotion. The run proves a binary expert recovery signal. It does not prove that token starvation was the only blocker, that the curve would reach 80%, or that the recovered model could behave coherently in free generation.

V1a took 4,477.2 seconds and used two passes over the same window schedule. Compared with V0, it changed token budget, data mix, and the nominal EOS objective together. That makes it useful as a receipt-hardening run, not a causal ablation. The small gain in pass two is ambiguous: it could mean the curve is flattening, the objective is wrong, or the run simply needs more matched tokens. V1b is the experiment that separates those explanations.

5.3 Free-running behavior did not recover

Both OLMoE receipts store three generation probes: a factual continuation, a Python function, and a tool-call prefix. At the binary floor, completions degenerated into repeated words or punctuation. After recovery, V0 still produced dashes and blank-line loops. V1a improved surface form but remained repetitive or schema-like rather than coherent. For example, the final tool-call probe repeatedly emitted weather-field fragments instead of a completed call.

This gap matters because BRQ is a behavior program. Teacher-forced top-1 and KL can recover ahead of autoregressive behavior. The correct conclusion is not that BRQ failed, and not that it succeeded. The conclusion is that BRQ needs an objective that directly budgets stopping, tool syntax, rare decision tokens, and free-running non-degeneracy.

Table 6: Generation-probe outcome summary.

Receipt	Free-running outcome
OLMoE floor	repeated common tokens, punctuation, or blank structures
OLMoE V0 final	dash and newline loops; no coherent completion
OLMoE V1a final	more recognizable schema fragments, still repetitive and incomplete
Qwen controls	repeated characters, numbers, punctuation, or short phrase loops

This is why the paper uses the word "recovery" carefully. BRQ recovered part of the teacher-forced distribution. It did not recover generation policy.

5.4 Dense controls: reservation was not the whole mechanism

The first interpretation of OLMoE was that reserving the spine caused the recovery. The Qwen3-8B controls corrected that reading. When all main linears were binarized, recovery rose from 0.0% top-1 to about 22% and then stalled. When attention and spine were reserved and only MLPs were binarized, the final result was still only 23.0%.

Table 7: Qwen3-8B binary controls.

Arm	Final top-1	Note
all linears	22.27%	partial, OOM after window 16–21
MLP only, spine reserved	22.95%	completed one pass

The revised reading is sharper: reservation is necessary for byte accounting and routing stability, but it is not what makes cheap recovery trainable. Expert granularity is the key variable. OLMoE's expert tensors are small enough that windowed recovery can move them. Qwen3-8B's dense matrices are much larger and stalled under the same budget.

5.5 All-in rate interpretation

The OLMoE ledger reports a binary expert rate of 1.125 bpw and an all-in rate of 2.15 bpw if the spine remains bf16, or 1.34 bpw if the spine is treated as MXFP4. Those rates are not release numbers. They are accounting checks that keep the probe honest about what would be stored if the trained latents were actually packed.

The K3 projection is more important strategically. Because K3 is overwhelmingly routed experts, a binary expert container plus MXFP4 spine projects to 1.191 bpw all-in. That is the reason the BRQ lane exists at

all. The result would be meaningless if it required a dense fp16 shadow of the model.

6 Interpretation

BRQ is not a new rounding formula. It is a change in what the low-bit problem is allowed to be. Static post-training quantization asks for the nearest useful code under a fixed representation. BRQ assumes that at one bit the nearest code is not the useful one. The code must be trained toward the teacher function.

The MoE structure makes that plausible because the expert tensors are independent enough to train locally after routing. With the router frozen, each expert behaves like a small MLP seen only on routed activations. K3 makes the arithmetic attractive because almost all parameters are routed experts while the spine is small. The OLMoE result supports the mechanism at small scale. The Qwen controls warn against generalizing it to dense binary models.

The most important negative result is the teacher-forced/free-running split. The recovered model can move toward the teacher distribution while still failing to generate usable outputs. This is exactly the failure mode that earlier BTL work already found in other low-bit settings. The next BRQ experiment must therefore promote on behavior, not on top-1 slope.

6.1 Why MoE expert granularity matters

An MoE layer routes each token to a small subset of experts. If the router remains fixed, each expert can be trained against the teacher activations it actually receives. This breaks the recovery problem into many smaller problems instead of one enormous dense backward pass. It also changes the data problem: common experts get many examples, rare experts may need usage-aware sampling or protected higher precision.

The Qwen control is the warning label. Holding attention and the rest of the spine did not lift the dense binary stall. That does not prove dense binary recovery is impossible. It proves BRQ's cheap small-run slope should be attributed to the MoE shape, not merely to the act of reserving some tensors.

6.2 Why behavior must remain the gate

The most seductive false claim would be: top-1 rose from 3.81% to 49.38%, so scaling will finish the job. The saved generations block that claim. A model can agree more often with the teacher under teacher forcing and still fail when it has to feed its own tokens back into the next step.

For BRQ, the behavior gate must include:

- ordinary instruction following;
- code completion and execution-backed repair;
- tool names, argument keys, delimiters, and malformed-output handling;
- explicit stop decisions and EOS behavior;
- false-premise rejection and abstention;
- repetition and degeneration checks.

Those categories are not decoration. They are the behaviors earlier low-bit runs lost first.

7 The V1b experiment that should run next

The V1a audit proposed a matched A/B because the existing run changed too many things at once.

7.1 Arm A: token-budget control

Arm A continues the V1a data and objective for two additional complete passes. It tests whether V1a was mainly token-starved. If the curve keeps climbing and generations improve, BRQ buys more training budget. If top-1 moves while generations remain broken, the objective needs to change.

7.2 Arm B: corrected objective

Arm B uses the same binary initialization, optimizer steps, token positions, window order, and evaluation checkpoints, but changes the objective. It must use an instruction-capable teacher or teacher-compatible chat template, train on a mixture of general text, code, tools, arguments, errors, and stop decisions, and allocate realized loss mass by target share:

- 8–12% to EOS and termination decisions;
- 12–18% to tool names, argument keys, delimiters, and error-recovery spans;
- the remainder to ordinary next-token KL.

The phrase "realized loss mass" is important. V1a recorded `eos_weight=4.0`, but did not show how much of the objective EOS actually occupied. If a batch has too few critical tokens, the training step should fail or resample instead of silently becoming uniform KL.

7.3 Promotion decision

V1b promotes only if one arm reaches at least 80% paired conditional retention, stored free-running cases are non-degenerate, and the result reproduces on a second seed. Anything short of that is still useful, but it remains a research result.

8 Promotion contract

The existing V1a receipt fails the promotion contract by design. A promoted BRQ result needs:

- exact model and tokenizer revisions;
- code hash, dependency versions, seed, hardware, and wall cost;
- binary initialization method and receipt hash;
- calibration source manifest, licenses, split, row IDs, and decontamination against evaluation;
- actual optimizer steps, token positions, skipped steps, and loss-mass allocation;
- paired teacher and student metrics on a train-disjoint frozen suite;

- at least 32 free-running cases during research, covering code, tools, ordinary instructions, and stopping;
- repetition, EOS, malformed-output, and manual-coherence results;
- packed artifact hash, physical bpw, file size, and runtime smoke before any release claim.

Scaling to a 35B or K3-class run requires at least 80% paired conditional retention, non-degenerate generations, and reproduction on a second seed. Until then, BRQ is a research program with a real signal, not a released compression method.

9 What this paper allows BTL to say

The current evidence supports the following public claims:

- BRQ is BTL's under-2-bit MoE recovery lane.
- Static post-training range selection did not reach one-bit behavior on BTL-4.
- On OLMoE-1B-7B, binary routed experts recovered from 3.81% to 49.38% teacher top-1 under hard-forward STE-KD.
- The recovered OLMoE receipts still failed free-running generation, so the run did not promote.
- A dense Qwen3-8B control stalled near 23%, supporting the view that expert granularity is the cheap-recovery regime.

The current evidence does not support these claims:

- BTL has a released one-bit or under-2-bit BRQ artifact.
- BRQ has achieved 80%, 90%, or 95% behavior retention.
- The V1a run proves that more tokens alone solve one-bit recovery.
- BRQ reproduces Bonsai's proprietary transformation.
- Teacher-forced top-1 is a substitute for code, tools, stopping, and free generation.

10 Limitations

Teacher-forced metric only. The strongest numerical gains are next-token agreement, KL, and perplexity on fixed chunks. They do not prove usable autoregressive behavior.

Tiny free-generation sample. Each receipt stores only three generation probes. The probes are enough to block promotion, not enough to characterize all failure modes.

No packed artifact. The saved latents are floating-point research state. No binary runtime file, bpw-verified artifact, or decode smoke was produced.

Initialization not fully tested. The V1a artifact did not receipt SparseGPT or LDLQ initialization. The paper cannot claim that lever as measured in BRQ V1a.

Confounded data/objective change. V1a changed more tokens, data mix, and EOS weighting together. The matched V1b A/B contract remains the right next test.

Proxy scale. OLMoE-1B-7B is a real MoE, but it is not BTL-4 and not K3. The result supports a scaling hypothesis. It does not price the final run.

11 Artifacts and reproducibility

The BRQ receipts and implementation are local to this repository:

- implementation: `btl-data-forge/training/modal_moe_1bit_recovery.py`
- OLMoE V0: `btl-data-forge/artifacts/moe1bit/olmoe-binary-2026-07-30.json`
- OLMoE V1a: `btl-data-forge/artifacts/moe1bit/olmoe-binary-v1a-2026-07-30.json`
- Qwen dense partial: `btl-data-forge/artifacts/moe1bit/qwen3-8b-binary-dense-PARTIAL-2026-07-30.json`
- Qwen MLP control: `btl-data-forge/artifacts/moe1bit/qwen3-8b-binary-mlp-spine-reserved-2026-07-30.json`
- recipe: `btl-data-forge/docs/one-bit-moe-recipe-2026-07-30.md`
- audit: `btl-data-forge/docs/brq-v1-audit-and-v1b-contract-2026-08-01.md`

These files are enough to inspect the reported runs and reproduce the local smoke path. They are not enough to reproduce a promoted one-bit artifact, because such an artifact does not yet exist.

12 Conclusion

BRQ changed the one-bit MoE question from static compression to recovery. The OLMoE experiments show that binary routed experts can relearn a large amount of teacher-forced structure from a near-dead floor. The Qwen controls show that this is not a generic dense binary rescue. The saved generations show that the result is not yet behaviorally usable.

That combination is the contribution. BRQ is the right research lane because it explains what post-training range selection cannot do, identifies expert granularity as the tractable unit, and sets the receipts required before any larger run is allowed. It is not yet a release paper. It is the paper that should prevent a premature release claim.

Appendix A: full OLMoE V0 curve

Table 8: Full V0 evaluation curve from `olmoe-binary-2026-07-30.json`.

Tag	Top-1	KL	PPL
floor	0.0381	7.3953	18446.0
after 0–2	0.3620	2.4309	104.5
after 2–4	0.4290	1.8245	57.5
after 4–6	0.4557	1.6411	47.6
after 6–8	0.4684	1.5740	44.2
after 8–10	0.4661	1.5435	43.0
after 10–12	0.4743	1.4750	40.2
after 12–14	0.4749	1.4617	39.4
after 14–15	0.4785	1.4555	39.1

The curve is why the research lane stayed alive: almost all improvement arrived before any behavior claim was

safe. The last row is still not a promoted model because generation remained degenerate.

Appendix B: full OLMoE V1a curve

Table 9: Full V1a evaluation curve from `olmoe-binary-v1a-2026-07-30.json`.

Tag	Top-1	KL	PPL
floor	0.0381	7.3953	18446.0
p0 0–2	0.3158	2.9159	168.3
p0 2–4	0.3919	2.0594	73.8
p0 4–6	0.4329	1.8250	57.3
p0 6–8	0.4424	1.7332	53.1
p0 8–10	0.4427	1.7098	51.8
p0 10–12	0.4401	1.6965	50.5
p0 12–14	0.4561	1.6561	49.1
p0 14–15	0.4613	1.6262	47.1
p1 0–2	0.4733	1.5820	44.9
p1 2–4	0.4834	1.5318	42.7
p1 4–6	0.4840	1.5138	41.9
p1 6–8	0.4801	1.4939	41.0
p1 8–10	0.4844	1.4958	41.0
p1 10–12	0.4837	1.4800	40.5
p1 12–14	0.4906	1.4663	40.0
p1 14–15	0.4938	1.4481	39.1

The second pass did not collapse, but it also did not change the conclusion. The run moved from 46.13% to 49.38% over pass two and kept the same free-running weakness.

Appendix C: Qwen control curves

Table 10: Qwen3-8B all-linear binary partial receipt.

Tag	Top-1	KL	PPL
floor	0.0000	17.4860	184879486
after 0–5	0.2152	4.2925	490.9
after 4–9	0.2243	4.0406	386.5
after 8–13	0.2223	4.0125	366.4
after 12–17	0.2327	3.9670	346.6
after 16–21	0.2227	3.9882	366.1

Table 11: Qwen3-8B MLP-only spine-reserved receipt.

Tag	Top-1	KL	PPL
floor	0.0000	23.1016	66630109642
after 0–5	0.1706	4.4762	590.4
after 4–9	0.2087	4.1206	413.9
after 8–13	0.2100	4.0550	382.1
after 12–17	0.2132	4.0018	361.3
after 16–21	0.2171	3.9657	361.6
after 20–25	0.2272	3.9196	338.1
after 24–29	0.2311	3.8785	334.0
after 28–33	0.2367	3.8877	337.5
after 32–35	0.2295	3.8986	350.5

These controls are imperfect, but they prevent an easy misreading. The OLMoE result should not be reduced to “reserve the spine.” Reserving dense attention and spine in Qwen3-8B did not unlock the same recovery slope.

Appendix D: minimum V1b receipt schema

A BRQ V1b receipt should be rejected if it cannot answer the following questions from the saved files alone:

- Which exact model and tokenizer revisions were loaded?
- What code hash, dependency versions, device type, seed, and wall cost produced the run?
- Which binary initialization was used, and what is its receipt hash?
- Which rows entered calibration, under which licenses and source IDs?
- Which evaluation rows were excluded from training by ID or content hash?
- How many optimizer steps ran, how many were skipped, and how many token positions were used?
- What share of realized loss mass went to EOS, tool names, argument keys, delimiters, and ordinary KL?
- What were the paired teacher and student metrics on a train-disjoint suite?
- What did at least 32 free-running generations look like, and how many were repetitive, malformed, or non-terminating?
- If a packed artifact exists, what are its hash, file size, physical bpw, tensor inventory, and runtime smoke result?

References

[1] Bad Theory Labs. *Behavior Before Perplexity: Compressing a 27B Agentic Model by Optimizing the Deployed Contract*. Technical report, 2026.

[2] Bad Theory Labs. *Range Before Representation: Behavior-gated two-bit quantization of a 35B mixture-of-experts model into a released 9.96 GB GGUF*. Technical report, 2026.

[3] Bad Theory Labs. *BTL Research Ledger*, 2026-08-18.

[4] Bad Theory Labs. *BRQ: Behaviour-Relearned Quantization, the 1-bit MoE recipe*, 2026-07-30.

[5] Bad Theory Labs. *BRQ V1 audit and V1b experiment contract*, 2026-08-01.

[6] Bad Theory Labs. *Has anyone uncovered Bonsai's quantization method?*, 2026-07-29.