PINPOINT QUANTIZATION: A MECHANISTIC ACCOUNT OF TAIL-CAPABILITY FAILURES IN QUANTIZED LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Weight quantization is essential for deploying large language models (LLMs), yet quantization below 4 bits can sharply degrade accuracy. Prior work has primarily characterized this degradation through model behavior or weight statistics, offering limited insight into its internal causes or how to enable targeted mitigation. Instead, we provide a *mechanistic account* of this failure by tracing capability-specific degradation to internal model components. We find that the damage is strikingly selective: it disproportionately impairs "tail" capabilities (*e.g.,* niche languages, long-context processing, complex factual queries, and multi-hop reasoning) while largely preserving "head" capabilities. To explain this selectivity, we analyze module function and discover two predominant perturbation mechanisms: attention heads undergo *semantic direction shifts*, whereas MLP neurons undergo *activation-magnitude suppression*. Crucially, the mechanistic fragility is concentrated in a sparse set of key modules: restoring their weight precision recovers most of the accuracy loss on the degraded tasks, suggesting that precise bit allocation, rather than bit width alone, is critical to sub-4-bit quantization. As a practical application of these findings, we introduce `Pinpoint Quantization`, which uses causal-effect scores to guide group-knapsack bit allocation. Across 18 benchmarks and dense and MoE models ranging from 4B to 235B parameters, it achieves near-lossless accuracy at an average weight precision below 4 bits.

1 INTRODUCTION

Large language models (LLMs) now scale from billions toward trillions of parameters (OpenAI, 2023; Team, 2024; DeepSeek-AI, 2025; Yang et al., 2025; Team, 2026), making weight quantization essential for practical deployment under real-world memory and latency constraints. In practice, although many carefully engineered low-bit designs (Kim et al., 2024; Liu et al., 2025; Shao et al., 2024; Huang et al., 2025) have pushed the compression ratio, current methods remain bottlenecked at 4 bits, with notable degradation at 2- and 3-bit precision (Kim et al., 2025b; Yan et al., 2026).

Prior studies have primarily characterized quantization-induced degradation through model behavior, such as changes in perplexity or benchmark accuracy (Egashira et al., 2024; Kim et al., 2025a; Li et al., 2025), or through weight statistics, such as outliers that motivate clipping and smoothing (Lin et al., 2024a; An et al., 2025; Shao et al., 2025). These empirical observations are valuable for describing ***what*** breaks, but provide limited insight into ***why*** it breaks. In particular, it remains unclear which internal computations quantization disrupts and whether these disruptions causally account for the observed failures. Meanwhile, mechanistic interpretability has reverse-engineered many internal computations of full-precision LLMs (Lindsey et al., 2025; Gurnee et al., 2026), yet has rarely examined quantized models, despite their widespread real-world deployment.

To bridge this gap, we provide a mechanistic account of low-bit quantization failure by tracing capability-specific degradation to internal modules (Figure 1). First, we find that quantization damage is strikingly selective (Section 3): quantization disproportionately degrades "tail" capabilities including niche languages, long-context processing, complex factual queries, and multi-hop reasoning, while largely preserving their "head" counterparts. Here, "tail" denotes these evaluated stress-test settings rather than a measured property of the training distribution. Then, to investigate the internal basis of this selectivity, we analyze the quantization effects on attention heads and MLP neurons (Section 4). Building on causal intervention techniques (Goldowsky-Dill et al., 2023; Wang et al.,

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.

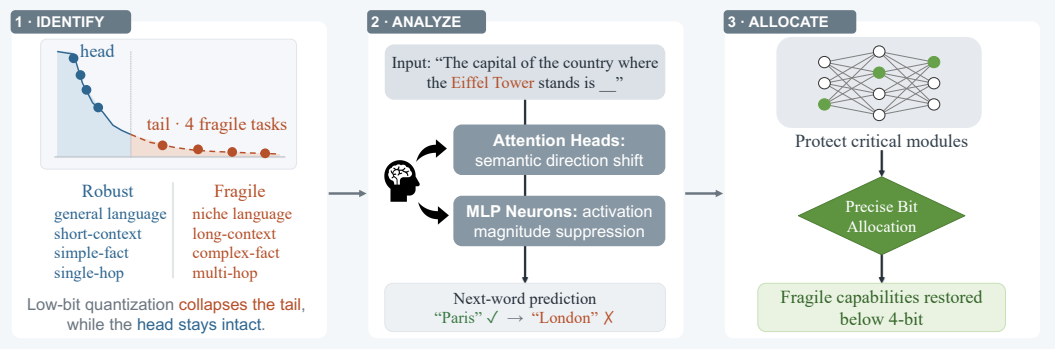


Figure 1: The study involves three parts: (1) *identify* the fragile capabilities, uncovering a head/tail divide in quantization damage; (2) *analyze* the underlying cause, revealing two human-understandable mechanisms; and (3) *allocate* precision to critical modules, restoring fragile capabilities below 4-bit.

2023), we introduce *quantization patching* to identify modules whose quantization contributes most to output errors. We then use *LogitLens* (nostalgebraist, 2020) to examine the semantic content of activations in these identified modules.

The analysis results reveal two predominant perturbation mechanisms: attention heads undergo a *semantic direction shift*, losing context while retaining signal strength, whereas MLP neurons undergo *activation magnitude suppression*, weakening signals while preserving semantics. These mechanisms link module-level disruptions to capability-specific failures. For niche languages, degradation is driven primarily by MLP activation suppression. By contrast, long-context processing degrades sharply when either attention or MLP modules are quantized. Crucially, this fragility is concentrated in a sparse subset of modules: restoring the weight precision of these modules from INT3 to FP16 recovers over 80% of the accuracy loss across the four fragile capabilities. Conversely, quantizing the identified modules in an otherwise FP16 model reproduces most of the loss. These bidirectional interventions show that the rankings identify modules with substantial effects on predictions, not merely large local numerical errors.

These findings suggest that precise bit allocation, rather than bit width alone, is critical to sub-4-bit quantization. The same causal-effect scoring principle used to localize quantization damage also quantifies module criticality, providing a direct signal for precision allocation. As a practical application, we introduce `Pinpoint Quantization` (Section 5): causal-effect scores serve as knapsack *values* and parameter counts as *costs*, and per-module bit-widths are solved under a global budget via *dynamic programming* (Cormen et al., 2022). Evaluated on 18 benchmarks across 6 domains, Pinpoint Quantization achieves near-lossless performance below 4 bits, spanning 4B–235B models and both Dense and Mixture-of-Experts architectures. Our allocation strategy is complementary to quantizers like GPTQ (Frantar et al., 2023): these methods determine *how* to quantize weights at a given bit width, whereas ours determines *where* to allocate precision.

Overall, our main contribution is a mechanistic study of quantization-induced capability loss. Beyond uneven degradation, we localize sensitive modules, characterize their output perturbations, and validate their criticality. We extend the intervention-based scoring principle to practical matrix-level precision allocation. The resulting "identify–analyze–allocate" pipeline offers a path toward more aggressive compression while mitigating capability degradation.

2 Related Work

Mechanistic Interpretability. Interpreting the internal mechanisms of LLMs has become increasingly urgent (Bereska & Gavves, 2024; Zhang et al., 2026a), motivating efforts to attribute model predictions to a sparse set of internal components. Activation patching (Vig et al., 2020; Meng et al., 2022) is one of the foundational techniques to this end: given a clean input and a minimally perturbed (corrupted) counterpart, it replaces the activation of a selected component with the corresponding corrupted activation, and measures the resulting change in the model's prediction as an estimate of that component's causal contribution. It has been used to localize behaviours ranging from indirect

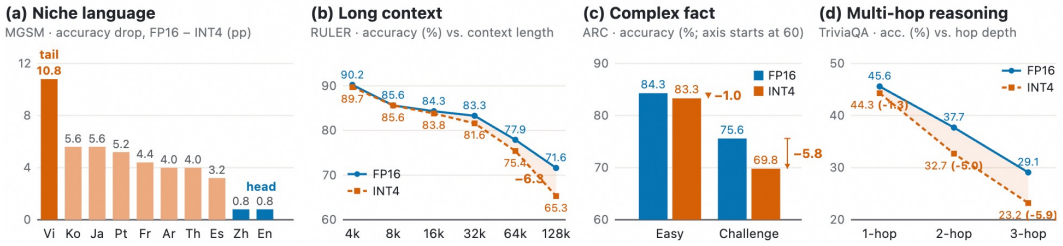


Figure 2: Evaluation of quantization-induced performance degradation on head and tail capabilities.

object identification (Wang et al., 2023) to arithmetic reasoning (Zhang et al., 2024) and answer binding (Lieberum et al., 2023). Complementarily, the LogitLens (nostalgebraist, 2020) decodes intermediate representations into the vocabulary space, providing a non-interventional view of how predictions are progressively refined across layers (Wiegreffe et al., 2025).

Post-Training Quantization for LLMs. Existing PTQ approaches (Zhao et al., 2025) fall into three families: compensation-based methods that use Hessian-guided weight updates (Frantar et al., 2023; Liu et al., 2024a; Zhang & Shrivastava, 2025; Guan et al., 2024), rotation-based methods that homogenize weight distributions via orthogonal transforms (Chee et al., 2023; Tseng et al., 2024; Liu et al., 2025; Ashkboos et al., 2024), and saliency-based methods that protect sensitive weights through scaling or smoothing (Lin et al., 2024b; Ma et al., 2024; Lee et al., 2024; Xiao et al., 2023). These techniques address *how* to quantize a weight tensor at a given bit-width but do not determine *where* to allocate precision across the model. Rather than applying a uniform bit-width, mixed-precision methods assign per-component precision by protecting outliers or task-sensitive weights at higher precision (Kim et al., 2024; Dettmers et al., 2024; Lee et al., 2024; Xiao et al., 2025), formulating precision assignment as constrained optimization over surrogate losses (Huang et al., 2025; Bai et al., 2025), or exploiting architectural priors such as MoE expert routing (Duanmu et al., 2025; Saxena et al., 2025; Zhao et al., 2024). Closely related, Zhang et al. (2026b) applied reasoning-behavior attribution to guide selective precision protection of reasoning models. We instead localize components by their quantization-induced output effects and characterize the resulting attention and MLP perturbations, focusing on quantization fragility rather than behavioral importance.

3 Identifying Quantization-Fragile Capabilities

The LLMs utilized in this work comprise the Qwen3-Instruct-2507 series (Yang et al., 2025) and Llama3-8B-Instruct (Team, 2024), spanning 4B, 8B, 30B-A3B, and 235B-A22B across dense and mixture-of-experts architectures. The analysis experiments are mainly reported on Qwen3-4B, while the results on other LLMs are reported in the appendix. We use a group-wise affine MinMax quantizer by default (Appendix D.4). We evaluate quantization impact using standard benchmarks via OpenCompass (Contributors, 2023), as shown in Figure 2:

Niche vs. General Language. Evaluated on MGSM (Shi et al., 2023), quantization severely penalizes low-resource languages while preserving the high-resource ones (Joshi et al., 2020). The accuracy drop for Vietnamese reaches about 10.8%, whereas English and Chinese degrade by less than 1%.

Long vs. Short Context. The performance gap on RULER (Hsieh et al., 2024) between FP16 and INT4 widens as the sequence length increases. The quantized model remains robust at short contexts (4k–8k) but suffers a sharp decline at lengths exceeding 32k tokens, indicating a loss of precision in representing distant dependencies.

Complex vs. Simple Fact. A "complexity gap" emerges in knowledge retrieval on ARC (Clark et al., 2018): the degradation on the *Challenge* subset is significantly larger than on the *Easy* subset, suggesting that quantization blurs the distinction between subtly different concepts.

Multi-hop vs. One-hop Reasoning: Logical depth is also sensitive to quantization noise. The accuracy drop on TriviaQA (Joshi et al., 2017) increases with the number of reasoning hops required, rising from 1.3% for 1-hop queries to 5.9% for 3-hop queries.

These results reveal a clear *head/tail divide* across representative, though non-exhaustive, capabilities. One possible explanation is that some tail capabilities depend on higher-order feature compositions

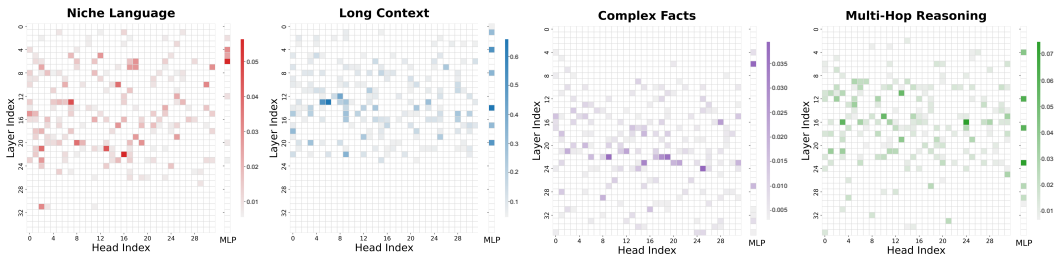


Figure 3: Causal-effect $\mathcal{E}(m)$ heatmaps of attention heads and MLP layers across four tail tasks.

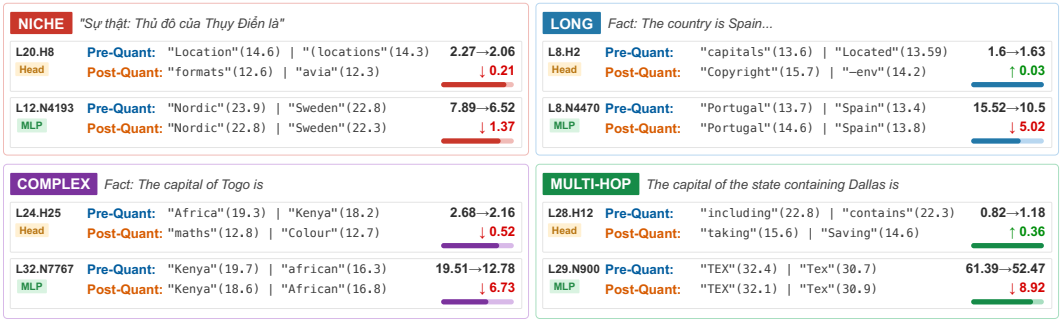


Figure 4: LogitLens decoding of key modules across four tail tasks, illustrating the changes in semantic directions (top-2 logits) and activation magnitudes (L_2) after quantization.

that are particularly sensitive to perturbations of their constituent features. This echoes the hierarchical-learning hypothesis discussed in Kumar et al. (2025); whether it explains the observed divide remains untested. Moreover, as illustrated in Figure 7 (Appendix A), the quantized model maintains accuracy on head tasks but fails across all four tail variants, characterized by a drastic suppression of the ground-truth token's probability and a flattening of the output distribution.

4 Internal Mechanisms of Quantization Fragility

4.1 Localizing Quantization-Sensitive Modules

Building on causal intervention techniques (Goldowsky-Dill et al., 2023; Wang et al., 2023), we introduce *quantization patching* to localize internal modules whose quantization perturbs model predictions. For each module m, we replace its FP16 activation with the output obtained by quantizing that module, while holding its input at the FP16 reference value and keeping all other modules in FP16. We denote this intervention by $do(m \leftarrow \mathrm{Quant}(m))$. We analyze attention heads individually. For MLPs, we first localize sensitive layers and then examine individual neurons within them.

Standard benchmark errors include both failures already present in FP16 and failures introduced by quantization. To isolate the latter, we construct a curated dataset of *prediction-flipped* instances: examples answered correctly by the FP16 model but incorrectly by its INT3 counterpart. These instances are generated from 20 templates covering the four tail scenarios (Appendix B). We split the dataset into a calibration set for module localization and a held-out test set for intervention-based validation. This conditional analysis characterizes quantization-induced failures within the selected scenarios, rather than estimating their prevalence on general benchmarks.

The causal effect $\mathcal{E}(m)$ is measured by the KL divergence between the FP16 output distribution and the distribution obtained when only module m is quantized: $\mathcal{E}(m) = D_{\mathrm{KL}}\big(\mathbb{P}_{\mathrm{fp}}(\cdot \mid x) \,\big\|\, \mathbb{P}^{(m)}(\cdot \mid x)\big)$, where $\mathbb{P}_{\mathrm{fp}}(\cdot \mid x) = \mathrm{softmax}(\mathbf{z}_{\mathrm{fp}})$ is the FP16 output distribution, and $\mathbb{P}^{(m)}(\cdot \mid x) = \mathrm{softmax}(\mathbf{z}^{(m)})$ is the distribution after $do(m \leftarrow \mathrm{Quant}(m))$. A larger $\mathcal{E}(m)$ indicates that quantizing m pushes the output distribution further from the FP16 reference; we regard high-scoring modules as *key modules*.

Figure 3 shows the resulting causal-effect maps and reveals two properties of quantization sensitivity. (1) **Sensitivity is concentrated in a small subset of modules.** Across the four tasks, fewer than

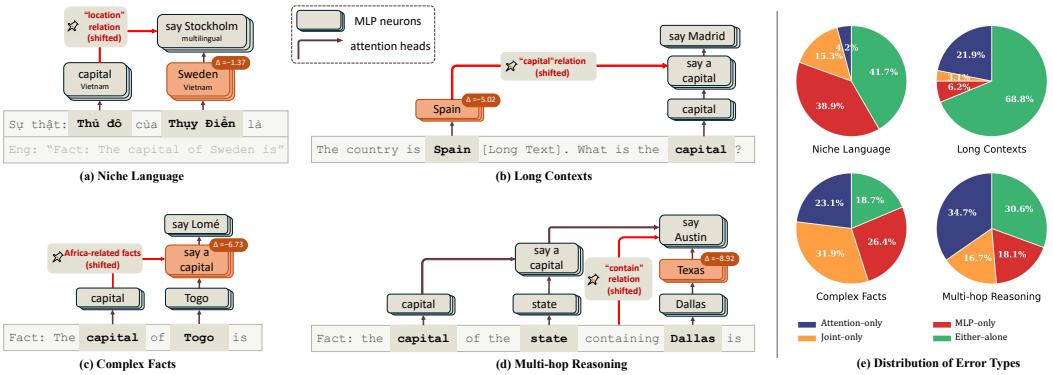


Figure 5: (a–d) The illustration of quantization suppressing MLP activation magnitudes (negative Δ) and shifting attention semantics (*e.g.,* "location", "contain"). (e) Four failure types defined by whether quantizing attention alone or MLPs alone is sufficient to induce an incorrect prediction.

10% of the analyzed modules account for most of the summed individual intervention scores. Thus, quantization does not perturb all modules equally: a small subset produces disproportionately large changes in the output distribution. (2) **The sensitive subset is task-dependent.** High-effect regions differ across capabilities: multi-hop reasoning is associated with a distinct set of deeper attention heads, whereas niche-language failures are associated with a separate cluster of MLP layers. The pairwise Jaccard overlap between task-specific module sets is below 30%, suggesting that a single shared ranking does not fully capture sensitivity across tasks.

4.2 Characterizing Module-Level Perturbations

Having localized the key modules, we next examine *how* quantization changes their contributions to model computation. Following the residual stream framework (Elhage et al., 2021), we represent each attention head and MLP neuron by its additive contribution to the residual stream. We use LogitLens (nostalgebraist, 2020) to approximately inspect the vocabulary-space content of these contributions.

Let $x \in \mathbb{R}^d$ denote the normalized representation at the position being analyzed. For attention head h, let K^h and V^h contain the keys and values at positions accessible under the causal mask. The residual-stream contribution of a head h and a gated MLP neuron j are as follows, respectively:

$$O^h(x) = \sigma\left(\frac{xW_Q^h(K^h)^\top}{\sqrt{d_h}}\right)V^h W_O^h, \quad \mathrm{MLP}^j(x) = \phi(xW_{\mathrm{gate}}^j)(xW_{\mathrm{up}}^j)W_{\mathrm{down}}^j. \tag{1}$$

Here, σ denotes the softmax function over key positions, and ϕ denotes the SiLU activation. W_{gate}^j and W_{up}^j are the corresponding projection columns, and W_{down}^j is the corresponding down-projection row. We quantify contribution magnitudes using $\|O^h(x)\|_2$ and $\|\mathrm{MLP}^j(x)\|_2$.

This analysis reveals two predominant perturbation mechanisms: (1) **Semantic direction shifts in attention heads.** Quantization changes the semantic content of key attention-head outputs without consistently suppressing their magnitudes. In Figure 4, for example, the top decoded token of `L28.H12` changes from "contains" to "Saving", while its output norm increases from 0.82 to 1.18. This example illustrates a disruption in decoded content rather than a loss of signal magnitude. We refer to this pattern as a *semantic direction shift*, using LogitLens decoding as an approximate indicator of the output's semantic content rather than a complete characterization of its function. (2) **Activation-magnitude suppression in MLP neurons.** Key MLP neurons often retain their top decoded tokens after quantization but contribute smaller-magnitude vectors to the residual stream. For example, `L29.N900` retains "TEX" as its top decoded token, while its contribution norm decreases from 61.39 to 52.47. We refer to this pattern as *activation-magnitude suppression*. A plausible consequence is that attenuated task-relevant contributions exert less influence relative to other residual-stream contributions. However, directly establishing this competition mechanism requires additional interventions and is beyond the scope of the present analysis.

Table 1: Cross-model mechanism consistency. Cos_{attn}: cosine distance between matched FP16/INT3 attention-head outputs. Δ_{MLP}: mean relative MLP output norm change. Rec./Deg.: fractions of the FP16–INT3 accuracy gap recovered/reproduced by restoring/degrading the identified modules.

Model	Arch.	Params	Cos_{attn}	Δ_{MLP} (%)	Rec.	Deg.
Qwen3-4B	Dense	4B	0.62	−27.3	80.9%	82.2%
Llama3-8B	Dense	8B	0.55	−24.5	84.2%	87.5%
Qwen3-30B-A3B	MoE	30B	0.66	−29.1	89.6%	92.3%
Qwen3-235B-A22B	MoE	235B	0.71	−31.8	91.3%	93.8%

Capability-specific failure patterns. To connect module-level observations to prediction failures, we classify prediction-flipped instances by whether attention-only and MLP-only quantization reproduce the flip, yielding four mutually exclusive categories (Figure 5(e)). (1) **Attention-only failure**: only attention quantization induces a flip. This pattern is most prevalent in multi-hop reasoning (34.7%), consistent with semantic direction shifts disrupting reasoning composition. (2) **MLP-only failure**: only MLP quantization induces a flip. It is most prevalent in niche languages (38.9%, versus 4.2% for attention-only failure), consistent with reduced contributions from task-relevant neurons. (3) **Joint-only failure**: neither intervention alone induces a flip, but their combination does. It is most prevalent in complex factual queries (31.9%), suggesting that combined perturbations cross the prediction boundary; retaining either module type in FP16 prevents the flip. (4) **Either-alone failure**: either intervention alone induces a flip. It dominates long-context processing (68.8%), where preserving only one module type is insufficient under this protocol. These failure modes are not confined to factual recall. On mathematical reasoning and code generation, repairing direction shifts and magnitude suppression increases the likelihood the model assigns to correct solutions and code, suggesting that these quantization-induced distortions also affect more complex tasks (Appendix C).

4.3 Validating Causal-Effect Scores

To substantiate our localization findings, we test whether high causal-effect scores reflect genuine contributions to capability degradation rather than artifacts of the scoring metric or calibration data. We validate causal-effect scores through held-out bidirectional interventions, cross-model comparisons, and alternative ranking criteria.

Let A_{fp} and A_{q} denote FP16 and INT3 accuracies, respectively. Let $A_{\text{restore}}(S)$ denote accuracy after restoring modules S to FP16 in the INT3 model, and $A_{\text{degrade}}(S)$ accuracy after quantizing only S to INT3 in the FP16 model. We measure the fractions of this gap recovered and reproduced:

$$\text{Rec}(S) = \frac{A_{\text{restore}}(S) - A_{\text{q}}}{A_{\text{fp}} - A_{\text{q}}}, \quad \text{Deg}(S) = \frac{A_{\text{fp}} - A_{\text{degrade}}(S)}{A_{\text{fp}} - A_{\text{q}}}. \tag{2}$$

Predictive validity under bidirectional interventions. We test whether modules ranked highly on the calibration set also drive accuracy changes on held-out examples. Restoring the identified key modules to FP16 recovers over 80% of the accuracy gap across the four evaluated capability categories; quantizing the same subset in FP16 reproduces most of the full-quantization loss (Figure 6). Thus, large single-module intervention scores identify subsets with substantial effects on held-out accuracy.

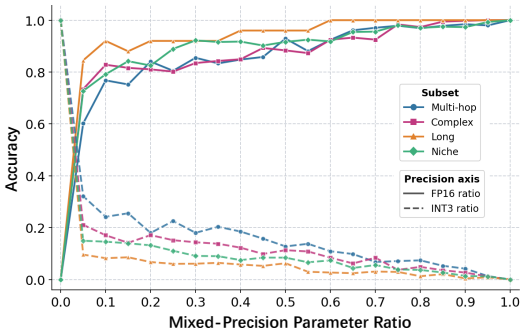


Figure 6: Mixed-precision quantization accuracy on four subsets. Solid: fraction kept in FP16; dashed: fraction quantized to INT3.

Generality across models and architectures. In Table 1, across four models spanning 4B–235B parameters and dense and MoE architectures, attention outputs exhibit directional changes and MLP outputs show magnitude suppression. Restoring the identified modules recovers an average of 86.5% of the FP16–INT3 accuracy gap, while degrading them reproduces 89.0%. These results support consistent perturbation patterns and concentrated, recoverable quantization sensitivity across the evaluated models.

Ranking effectiveness against alternatives. We compare causal-effect scores with random selection, weight and activation norms, and Hessian-diagonal scores. For each ranking, we evaluate two interventions under the same module-count budget. In Table 2, causal-effect ranking recovers 80.9% of the FP16–INT3 accuracy gap, compared with 70.2% for the Hessian-based ranking, and 44.7% for activation norms. It also reproduces the largest fraction of the gap when the selected modules alone are quantized (82.2%). Thus, the same intervention-based signal supports both localization and targeted recovery. Unlike magnitude-based criteria, the KL score measures the downstream distributional effect of quantizing a module and can therefore capture perturbations that change output content. The comparison supports its practical utility as a ranking signal.

Table 2: Comparison of ranking metrics.

Ranking metric	**Rec.**	**Deg.**
Random-10%	10.2%	12.8%
Weight ℓ_2 norm	20.4%	24.1%
Activation ℓ_2 norm	44.7%	38.5%
Hessian diag($\mathbf{H}$)	70.2%	73.8%
Causal $\mathcal{E}(m)$ (Ours)	**80.9%**	**82.2%**

5 Application: Pinpoint Quantization

The mechanistic findings motivate selective precision protection, but do not prescribe a universal module-wise allocation. For each target model and calibration distribution, we reuse the intervention-based scoring principle, recomputing intervention scores at hardware-friendly matrix granularity. The method has two stages: (i) *causal-effect scoring*, which measures the bit-dependent output perturbation induced by quantizing each weight matrix individually; and (ii) *budgeted bit allocation*, which selects one bit width per matrix using group-knapsack dynamic programming. Algorithm 1 summarizes the procedure. More implementation details are in Appendix D.

5.1 Causal Effect Scoring

Setup. Let $f(x;\theta)$ be an FP16 LLM with N quantizable weight matrices $\{W_i\}_{i=1}^N$, and let $\mathcal{B}$ be the set of candidate bit widths. We sample a calibration corpus $\mathcal{D}$ from general pretraining-style data.

Bit-dependent causal-effect scores. For each matrix W_i and candidate bit width b, we construct $f_{i\leftarrow b}$ by quantizing only W_i while retaining all other parameters in FP16. Let $p_{\text{fp}}(\cdot \mid x_{<t})$ and $p_{i,b}(\cdot \mid x_{<t})$ denote the full-vocabulary next-token distributions of the reference and perturbed models. We define the effect score as:

$$s_i(b) = \frac{1}{Z} \sum_{x\in\mathcal{D}} \sum_{t\in\mathcal{T}(x)} D_{\text{KL}}(p_{\text{fp}}(\cdot \mid x_{<t}) \,\|\, p_{i,b}(\cdot \mid x_{<t})) \,, \tag{3}$$

where $\mathcal{T}(x)$ contains the valid prediction positions in x and $Z = \sum_{x\in\mathcal{D}} |\mathcal{T}(x)|$. A larger $s_i(b)$ indicates that assigning b bits to matrix W_i causes a larger departure from the FP16 output distribution.

5.2 Budgeted Bit Allocation

Given importance scores $\{s_i(b)\}$, a bit-width $\pi(i) \in \mathcal{B}$ is selected for each module under a global weight-bit budget. Let n_i be the number of parameters in module i. Assigning bit-width b to module i incurs weight-bit cost $c_i(b) = n_i \cdot b$, and the total capacity constraint is $\sum_{i=1}^N c_i(\pi(i)) \le C$.

Optimization objective. The predicted total perturbation is minimized:

$$\min_{\pi} \sum_{i=1}^{N} s_i(\pi(i)) \quad \text{s.t.} \quad \sum_{i=1}^{N} c_i(\pi(i)) \le C, \;\; \pi(i) \in \mathcal{B}. \tag{4}$$

This is a *group knapsack* problem: each module i is a group, and exactly one item (bit-width) is chosen from each group.

Dynamic programming and quantization. We solve the group-knapsack formulation using dynamic programming and recover the selected bit widths by backtracking. The budget representation and recurrence are detailed in Appendix D.2. After allocation, each matrix is quantized using its assigned bit width. The allocator is complementary to the underlying quantizer. For inference, we extend GEMM kernels (Frantar et al., 2025) to support the selected bit widths (Appendix D.3).

Table 3: We compare the FP16 baseline, uniform 4-bit quantization, and Pinpoint Quantization (PinQ). Blue shading indicates our mixed-precision results, with parenthetical values showing the gap relative to FP16; subscripts on PinQ denote the average bit-width.

		Qwen3-235B-A22B			Qwen3-30B-A3B			Qwen3-4B		
Domain	**Benchmark**	**FP16**	**INT4**	**PinQ$_{3.3}$**	**FP16**	**INT4**	**PinQ$_{3.5}$**	**FP16**	**INT4**	**PinQ$_{3.8}$**
General	MMLU	88.7	88.2	88.6 (-0.1)	85.0	84.4	84.1 (-0.9)	74.8	72.2	74.0 (-0.8)
	ARC-Challenge	86.1	84.3	85.1 (-1.0)	83.1	82.5	84.5 (+1.4)	75.6	69.8	75.0 (-0.6)
	ARC-Easy	92.8	92.0	91.8 (-1.0)	89.4	89.1	90.0 (+0.6)	84.3	83.3	84.5 (+0.2)
	GPQA-Diamond	60.3	57.3	62.6 (+2.3)	56.2	54.2	59.4 (+3.2)	36.9	34.2	35.3 (-1.6)
Alignment	AlpacaEval-2.0	89.4	86.6	87.5 (-1.9)	68.5	68.7	68.6 (+0.1)	59.3	58.2	58.8 (-0.5)
	IFEval	88.5	87.1	88.2 (-0.3)	84.1	81.2	82.1 (-2.0)	87.2	87.0	87.3 (+0.1)
Math & Reason.	GSM8k	91.0	88.7	90.0 (-1.0)	89.8	88.8	89.7 (-0.2)	89.8	87.6	88.9 (-0.9)
	TriviaQA	76.5	73.8	75.9 (-0.6)	63.7	60.7	61.8 (-1.9)	41.9	39.1	40.2 (-1.7)
	WinoGrande	80.0	79.6	79.9 (-0.1)	77.0	76.1	76.7 (-0.3)	66.8	64.7	65.8 (-1.0)
Agent & Coding	LiveCodeBench	73.2	72.5	74.5 (+1.3)	48.5	47.9	48.3 (-0.2)	58.5	52.2	57.8 (-0.7)
	HumanEval	94.6	94.2	94.8 (+0.2)	93.8	92.0	91.4 (-2.4)	87.2	83.5	87.2 (+0.0)
	T-Eval	90.7	90.7	90.6 (-0.1)	88.9	87.0	88.7 (-0.2)	88.8	87.6	88.6 (-0.2)
Multilingual	BELEBELE	80.5	79.4	79.9 (-0.6)	75.6	74.9	75.9 (+0.3)	66.5	62.4	66.2 (-0.3)
	MGSM	77.4	76.3	78.4 (+1.0)	70.4	72.0	71.6 (+1.2)	62.0	58.0	60.5 (-1.5)
	Multi-IF	83.9	83.0	84.0 (+0.1)	77.7	76.2	77.2 (-0.5)	77.4	73.4	76.8 (-0.6)
	MMMLU	84.7	83.8	84.5 (-0.2)	80.2	79.4	79.7 (-0.5)	69.2	66.5	68.5 (-0.7)
Long Context	LongBenchV2	35.0	34.3	33.0 (-2.0)	30.2	29.8	30.4 (+0.2)	20.5	18.9	20.7 (+0.2)
	RULER-128k	61.6	61.6	61.7 (+0.1)	58.4	57.9	58.6 (+0.2)	71.6	65.3	70.3 (-1.3)
Average (6 domains, 18 benchmarks)		79.7	78.5	79.5 (-0.2)	73.4	72.4	73.3 (-0.1)	67.7	64.7	67.0 (-0.7)

6 Evaluation of Pinpoint Quantization

Our mechanistic analysis motivates a practical question: can selective precision allocation preserve model quality more effectively than uniformly increasing precision? We evaluate Pinpoint Quantization (**PinQ**) to test this principle and identify the design choices that make it effective. We primarily evaluate the Qwen3 family (Yang et al., 2025), with additional baseline comparisons on Llama3-8B-Instruct. All quantized models are constructed from FP16 references and evaluated via OpenCompass (Contributors, 2023) on NVIDIA A100 80GB GPUs (Appendix E).

Downstream accuracy: selective precision pays off. PinQ translates intervention-guided allocation into a favorable accuracy–precision trade-off. Across the 18 benchmarks in Table 3, it outperforms uniform INT4 in mean score on all three Qwen3 models while using only 3.3, 3.5, and 3.8 average bits, respectively. Thus, retaining aggregate quality does not require assigning the same precision to every matrix. This benefit also appears in a capability identified as fragile in our analysis. On Qwen3-4B, PinQ raises RULER-128k from 65.3% under uniform INT4 to 70.3%, close to FP16's 71.6%. Preservation is nevertheless task-dependent: Qwen3-235B-A22B drops from 35.0% to 33.0% on LongBenchV2. The results therefore support targeted allocation as a practical compression strategy, not a guarantee of lossless performance on every capability.

In Table 4, PinQ also remains competitive at a more aggressive budget on Llama3-8B-Instruct. At comparable reported bit budgets, PinQ achieves higher average accuracy than the mixed-precision baselines, including SliM-LLM (Huang et al., 2025), SqueezeLLM (Kim et al., 2024), HIGGS (Malinovskii et al., 2025), HQQ (Badri & Shaji, 2023), AQLM (Egiazarian et al., 2024), TACQ (Xiao et al., 2025) and SpQR (Dettmers et al., 2024). This validates that effective compression requires not only flexibility in bit widths, but also a precise criterion for allocating them. PinQ provides such a criterion by measuring how quantizing each module changes the model's output distribution, linking precision decisions to their downstream consequences. Together, these results position mechanistic analysis as more than an explanation of quantization failure: it becomes an actionable basis for preserving model quality under a constrained precision budget.

Table 4: Comparison with low-bit baselines on Llama3-8B-Instruct. Bold indicates the best score among quantized configurations. Blue highlights the results of our PinQ.

Method	Avg. bits	Mixed	MMLU	GSM8K	ARC-E	ARC-C	WinoGrande	Average
FP16	16	✗	66.51	73.46	81.02	52.99	78.67	70.53
GPTQ	3.00	✗	56.30	51.96	74.90	42.47	74.58	60.04
HQQ	3.25	✓	55.23	47.98	77.04	42.98	68.55	58.36
AQLM	3.27	✓	58.72	50.87	77.10	45.90	76.26	61.77
SqueezeLLM	3.15	✓	59.66	49.99	77.12	45.93	76.27	61.79
HIGGS	3.25	✓	61.77	50.75	78.97	47.60	70.33	61.88
SpQR	3.14	✓	58.65	54.99	76.45	44.89	75.06	62.01
SliM-LLM	3.125	✓	61.95	66.24	78.63	45.29	75.74	65.57
TACQ	3.12	✓	61.97	67.07	**79.38**	46.03	72.33	65.36
PinQ	3.10	✓	**63.89**	**70.22**	78.92	**48.94**	**77.83**	**67.96**

Table 5: Ablation studies on the allocation pipeline on Qwen3-4B: (a) quantizer compatibility at a matched 4.0-bit budget; (b) planner choice; and (c) candidate bit set $\mathcal{B}$. Perplexity (PPL) is evaluated on C4. Blue highlights the default configuration in each subtable.

(a) **Quantizer Compatibility**

Quantizer	MMLU	PPL
GPTQ	73.92	16.17
+ PinQ	**74.62** (+0.70)	**15.93** (−0.24)
MinMax	72.15	17.12
+ PinQ	74.19 (+2.04)	16.04 (−1.08)

(b) **Planner Choice**

Planner	bits	MMLU	PPL
Greedy	3.0	60.39	20.96
DP	3.0	**65.73**	**19.35**
Greedy	4.0	73.28	16.89
DP	4.0	**74.19**	**16.04**

(c) **Bit Set**

$\mathcal{B}$	MMLU	PPL
2,4	72.15	17.12
2,4,8	73.48	16.38
2,3,4,8	73.85	16.07
2,3,4,6,8	**74.19**	**16.04**

Ablation studies: allocation complements quantization. PinQ's gains extend across quantizers, showing that precision allocation is a complementary design choice rather than a quantizer-specific refinement. At a matched 4.0-bit budget, PinQ improves MMLU by 0.70 and 2.04 points with GPTQ and MinMax, respectively, while reducing C4 perplexity in every case (Table 5a). Effective allocation therefore remains valuable even with a simple local quantizer: deciding where to spend precision complements how weights are quantized. The allocation strategy becomes particularly consequential under tight budgets. DP's advantage over greedy allocation grows from 0.91 MMLU points at 4.0 bits to 5.34 points at 3.0 bits (Table 5b), while expanding the candidate set favors flexible allocation under a global constraint, especially when precision budget is limited (Table 5c).

Discussion and Scope. The observed attention and MLP perturbation patterns characterize the measured output changes but do not explicitly determine bit assignments. Accordingly, PinQ is a practical adaptation of the diagnostic study, not a mechanism-specific allocation rule. Downstream results support this adaptation. Dynamic programming optimizes the additive score surrogate exactly, but does not account for all cross-module interactions. The effectiveness of the resulting allocation is therefore supported by downstream evaluation, not by the optimization guarantee alone. Module-level restoration establishes causal relevance, but does not uniquely attribute recovery to direction shifts or magnitude suppression. We therefore interpret these as predominant perturbation mechanisms, rather than an exhaustive causal decomposition of quantization failure.

7 Conclusion

This work connects selective capability degradation under quantization to sparse, causally consequential modules and characterizes distinct perturbation patterns in attention and MLP computations. Guided by intervention-based sensitivity scores, Pinpoint Quantization achieves near-FP16 aggregate performance at an average precision below 4 bits across the evaluated models.

More broadly, our findings suggest that mechanistic analysis can inform not only where to allocate precision, but also how to calibrate and correct quantized computations. Future work could explore calibration that better covers fragile capabilities and corrections targeting attention direction shifts and MLP magnitude suppression. These directions could extend mechanistic insights beyond bit allocation to targeted capability preservation.

AI USE STATEMENT

We used generative AI tools to assist with translation, language editing, and interpretation of results. We did not use generative AI tools to generate synthetic datasets or to clean or reformat experimental data.

The authors reviewed and revised the AI-assisted content and evaluated all interpretations and claims against the experimental evidence. We take full responsibility for the final manuscript, including its analyses, claims, and conclusions.

ETHICS STATEMENT

This work studies weight quantization and capability preservation in publicly available large language models using existing datasets and benchmarks. It does not involve recruiting human participants. Our findings motivate capability-specific evaluation rather than reliance on aggregate accuracy alone. Quantized models should undergo application-specific validation before deployment, particularly in high-stakes settings.

REPRODUCIBILITY STATEMENT

The main text and appendices document the experimental procedures and implementation choices to support reproducibility. Section 3 identifies the evaluated models and capability benchmarks, while Section 4 defines the quantization-patching protocol, causal-effect metric, and intervention-based validation measures. Appendices B and C describe the diagnostic data construction, calibration/test split, and selective-quantization error taxonomy. Section 5 and Algorithm 1 present the allocation procedure, with implementation details provided in Appendix D. Appendix E specifies the quantization settings, calibration mixture, and precision-accounting conventions. Section 6 and Appendix F report the evaluation settings, detailed allocation results, and inference-efficiency measurements.

REFERENCES

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Systematic outliers in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=rLX7Vyyzus.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dfqsW38v1X.

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, November 2023. URL https://mobiusml.github.io/hqq_blog/.

Runsheng Bai, Bo Liu, and qiang liu. SKIM: Any-bit quantization pushing the limits of post-training quantization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JpoDVFYx2w.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022. doi: 10.48550/ARXIV.2204.05862. URL https://doi.org/10.48550/arXiv.2204.05862.

Leonard F. Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - A review. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=ePUVetPKu6.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xrk9g5vcXR.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556, 2025. doi: 10.48550/ARXIV.2512.02556. URL https://doi.org/10.48550/arXiv.2512.02556.

Tim Dettmers, Ruslan A Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless llm weight compression. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Q1u25ahSuy.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 3029–3051. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.183. URL https://doi.org/10.18653/v1/2023.emnlp-main.183.

Haojie Duanmu, Xiuhong Li, Zhihang Yuan, Size Zheng, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. Mxmoe: Mixed-precision quantization for moe with accuracy and performance co-design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=pXoZLGMNDm&noteId=vmMvwkSQDg.

Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting llm quantization. *Advances in Neural Information Processing Systems*, 37:41709–41732, 2024.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12284–12303. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/egiazarian24a.html.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Elias Frantar, Roberto L. Castro, Jiale Chen, Torsten Hoefler, and Dan Alistarh. Marlin: Mixed-precision auto-regressive parallel inference on large language models. In *PPoPP*, pp. 239–251, 2025. URL https://doi.org/10.1145/3710848.3710871.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *CoRR*, abs/2304.05969, 2023. doi: 10.48550/ARXIV.2304.05969. URL https://doi.org/10.48550/arXiv.2304.05969.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. Aptq: Attention-aware post-training mixed-precision quantization for large language models. In *DAC*, pp. 107:1–107:6, 2024. URL https://doi.org/10.1145/3649329.3658498.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: what's the real context size of your long-context language models? *CoRR*, abs/2404.06654, 2024. doi: 10.48550/ARXIV.2404.06654. URL https://doi.org/10.48550/arXiv.2404.06654.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and XIAOJUAN QI. Slim-LLM: Salience-driven mixed-precision quantization for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=PO9bBEPNWy.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 6282–6293. Association for Computational Linguistics, 2020.

Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens JS Schaefer, Deokjae Lee, Yeonhong Park, Jae W Lee, and Hyun Oh Song. Guidedquant: Large language model quantization via exploiting end loss guidance. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=ZawsPjlIGu.

Junhan Kim, Ho-young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. Boa: Attention-aware post-training quantization without backpropagation. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=Uvj6XcSJ5d.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-Sparse Quantization, June 2024. URL http://arxiv.org/abs/2306.07629. arXiv:2306.07629 [cs].

Tanishq Kumar, Zachary Ankner, Benjamin Frederick Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=wg1PCg3CUP.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *AAAI*, pp. 13355–13364, 2024. URL https://doi.org/10.1609/aaai.v38i12.29237.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro,

Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason T. Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder: may the source be with you! *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=KoFOg41haE.

Zhen Li, Yupeng Su, Runming Yang, Zhongwei Xie, Ngai Wong, and Hongxia Yang. Quantization meets reasoning: Exploring LLM low-bit quantization degradation for mathematical reasoning. *CoRR*, abs/2501.03035, 2025. doi: 10.48550/ARXIV.2501.03035. URL https://doi.org/10.48550/arXiv.2501.03035.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *CoRR*, abs/2307.09458, 2023.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37:87766–87800, 2024a.

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. *Proceedings of machine learning and systems*, 6: 87–100, 2024b.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *EMNLP*, pp. 8181–8196, 2024a. URL https://aclanthology.org/2024.emnlp-main.467.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ogO6DGE6FZ.

Zuxin Liu, Thai Quoc Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh R N, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. APIGen: Automated PIpeline for generating verifiable and diverse function-calling datasets. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=Jfg3vw2bjx.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Affinequant: Affine transformation quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=of2rhALq8l.

Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtárik, and Dan Alistarh. HIGGS: pushing the limits of large language model quantization via the linearity theorem. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long*

Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025, pp. 10857–10886. Association for Computational Linguistics, 2025. URL https://doi.org/10.18653/v1/2025.naacl-long.543.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, 2025.

nostalgebraist. Interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020. LessWrong.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jKHmjlpViu.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=4qIP1sXcR1.

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 *ICLR*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. Dartquant: Efficient rotational distribution calibration for llm quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=LfcfwlLCHM.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. 2023.

Kimi Team. Kimi K3: open frontier intelligence. *CoRR*, abs/2607.24653, 2026. doi: 10.48550/ARXIV.2607.24653. URL https://doi.org/10.48550/arXiv.2607.24653.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip #: Even better llm quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning*, pp. 48630–48656. PMLR, 2024.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Sarah Wiegreffe, Oyvind Tafjord, Yonatan Belinkov, Hannaneh Hajishirzi, and Ashish Sabharwal. Answer, assemble, ace: Understanding how lms answer multiple choice questions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=6NNA0MxhCH.

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 *Proceedings of the 40th International Conference on Machine Learning*, pp. 38087–38099. PMLR, July 2023. URL https://proceedings.mlr.press/v202/xiao23c.html. ISSN: 2640-3498.

Hanqi Xiao, Yi-Lin Sung, Elias Stengel-Eskin, and Mohit Bansal. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=a201nfn3xX.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 483–498. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.41. URL https://doi.org/10.18653/v1/2021.naacl-main.41.

Xianglong Yan, ChengZhu Bao, Zhiteng Li, Tianao Zhang, Shaoqiu Zhang, Ruobing Xie, Samm Sun, and Yulun Zhang. D^2quant: Accurate low-bit post-training weight quantization for llms. *arXiv preprint arXiv:2602.02546*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Hengyuan Zhang, Zhihao Zhang, Ercong Nie, Mingyang Wang, Zunhai Su, Yiwei Wang, Qianli Wang, Shuzhou Yuan, Xufeng Duan, Qibo Xue, Zeping Yu, Chenming Shang, Xiao Liang, Jing Xiong, Hui Shen, Chaofan Tao, Zhengwu Liu, Senjie Jin, Zhiheng Xi, Dongdong Zhang, Sophia Ananiadou, Tao Gui, Ruobing Xie, Hayden Kwok-Hay So, Hinrich Schuetze, Xuanjing Huang, Qi Zhang, and Ngai Wong. Locate, steer, and improve: A practical survey of actionable mechanistic interpretability in large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10317–10362, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.502. URL https://aclanthology.org/2026.findings-acl.502/.

Nan Zhang, Eugene Kwek, Yusen Zhang, Hieu Nguyen, Prasenjit Mitra, and Rui Zhang. When reasoning meets compression: Understanding the effects of llms compression on large reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 62680–62706, 2026b.

Tianyi Zhang and Anshumali Shrivastava. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ISqx8giekS.

Wei Zhang, Chaoqun Wan, Yonggang Zhang, Yiu-ming Cheung, Xinmei Tian, Xu Shen, and Jieping Ye. Interpreting and improving large language models in arithmetic calculation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CfOtiepP8s.

Jiaqi Zhao, Ming Wang, Miao Zhang, Yuzhang Shang, Xuebo Liu, Yaowei Wang, Min Zhang, and Liqiang Nie. Benchmarking post-training quantization in llms: Comprehensive taxonomy, unified evaluation, and comparative analysis, 2025. URL https://arxiv.org/abs/2502.13178.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024.

APPENDIX OVERVIEW

The appendix presents quantization failure cases, diagnostic data construction, and selective-intervention analyses, followed by details of PinQ and its implementation, bit allocations, and inference efficiency. The overview below links to each section and subsection.

Contents		Page
I. Failure Cases and Diagnostic Analysis		
A. Examples of Quantization-Induced Failures		18
A.1	Tail-Language Instruction Following Failure (MGSM-ja)	18
A.2	Impact of Quantization on General Knowledge (ARC-Easy vs. Challenge)	18
A.3	Probability Distribution Shift under Quantization	19
B. Diagnostic Templates and Data Construction		19
C. Error Taxonomy under Selective Quantization		20
II. Quantization Method and Implementation		
D. Pinpoint Quantization Details		21
D.1	Accelerating Intervention Scoring with LayerCache	22
D.2	Dynamic Programming for Bit Allocation	23
D.3	Inference Kernels	24
D.4	Group-Wise MinMax Quantization	24
E. Implementation Details		25
III. Bit Allocation and Inference Efficiency		
F. Detailed Results		25
F.1	Bit Assignment Results	25
F.2	Inference Efficiency	26

A EXAMPLES OF QUANTIZATION-INDUCED FAILURES

This section provides illustrative INT4 failure cases and supplementary benchmark results. Evaluations use OpenCompass (Contributors, 2023) with temperature 0.7, top-p 0.8, top-k 20, and presence penalty 1.5. The examples characterize observed output errors; they do not by themselves establish their internal causes.

A.1 TAIL-LANGUAGE INSTRUCTION FOLLOWING FAILURE (MGSM-JA)

Task requirement. The model must output the final line in the exact format "答えは <integer>" and *not* append any extra text.

MGSM-ja

Prompt (Japanese). Solve this math problem. Give the reasoning steps before giving the final answer on the last line by itself in the format of "答えは ". Do not add anything other than the integer answer after "答えは ".
ジョンは時速60マイルで3時間車を運転した後に、自宅に非常に大事なものを忘れたことに気づいたため引き返します。4時間以内に自宅に戻ろうとしますが、最初の2時間は大渋滞で動けません。次の30分間を時速30マイルで運転し、その後時速80マイルで自宅に向かいました。4時間が経過したタイミングで、ジョンは自宅から何マイル離れていることになりますか？
Gold. 45
INT4 output (incorrect). . . .
1. John covers a distance of 60 x 3 = 180 miles from home.
2. After 2 hours of traffic (0 miles), he drives 30 mph for 0.5 hours: 30 x 0.5 = 15 miles.
3. **Error:** Since he has 4 hours total, and 2 hours were spent in traffic, he has 2 hours left (Should be $4 - 2 - 0.5 = 1.5$ hours).
4. **Error:** In total, John covers a distance of 180 + 15 + 160 = 355 miles.
(Logic Error: The model **added** the distances instead of **subtracting** from the 180-mile gap to home.)
. . .
"So, John is 355 miles away from home when 4 hours have passed."

What fails. The reasoning contains multiple inconsistent steps and produces a wrong final integer, also not matching the required prefix. This type of error is more frequent in tail languages and long token sequences, where quantization noise can accumulate and destabilize intermediate state tracking.

A.2 IMPACT OF QUANTIZATION ON GENERAL KNOWLEDGE (ARC-EASY VS. CHALLENGE)

We analyze the performance trade-offs between FP16 and INT4 models across different reasoning depths using the ARC datasets. Table 6 summarizes the accuracy and the resulting performance gap:

Table 6: Comparison of model performance on the ARC datasets before and after INT4 quantization.

Dataset	Evaluation	FP16 Baseline	INT4 Quantized	Δ (Gap)
ARC-Challenge	Accuracy	75.59%	69.83%	**−5.76%**
	Samples (Correct/Total)	223 / 295	206 / 295	−17
ARC-Easy	Accuracy	84.30%	83.25%	**−1.05%**
	Samples (Correct/Total)	478 / 567	472 / 567	−6

Observed failure pattern in ARC-c. The significant degradation (-5.76%) in the Challenge set indicates that tasks involving **scientific reasoning, multi-step logic, and strong distractors** are highly sensitive to weight precision. Under INT4 quantization, the model exhibits a weakened ability to integrate multiple factors, often falling for lexically plausible but logically inconsistent distractors.

Observed robustness in ARC-e. In contrast, the performance gap in the Easy set is minimal (-1.05%), where INT4 results nearly match FP16. This demonstrates that **basic commonsense recall**

and simple pattern matching are remarkably robust to precision reduction, as these tasks rely more on high-frequency factual retrieval than complex logical synthesis.

Summary. The "precision tax" of quantization is non-uniform: it is negligible for factual recall but becomes a bottleneck for high-order reasoning that requires fine-grained weight differentiation.

A.3 Probability Distribution Shift under Quantization

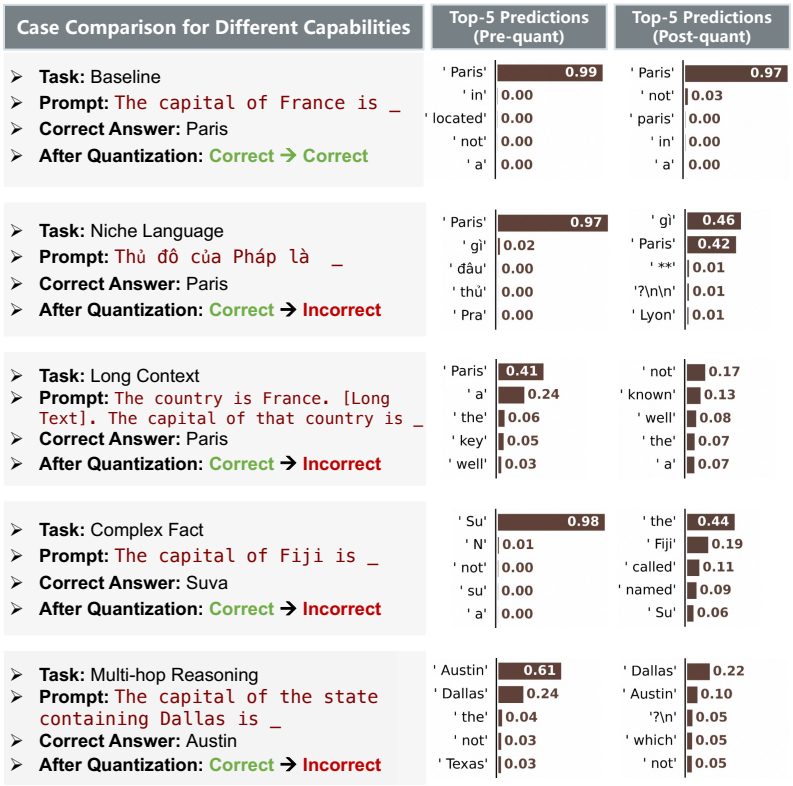


Figure 7: **Case Studies of Quantization Impact.** For the baseline task (*e.g.,* simple factual recall), the correct token's prediction probability remains dominant ($0.99 \rightarrow 0.97$). For the four tail tasks (*e.g.,* niche language), the probability distribution flattens or shifts incorrectly after quantization ($0.97 \rightarrow 0.42$).

Figure 7 presents probability-level case studies of quantization impact. Derived from a simple factual recall sample, the four tail versions serve as stress tests for the model's robustness. While the pre-quantized model correctly answers all varied prompts with high confidence, the quantized counterpart exhibits a sharp performance divergence: it maintains accuracy on the baseline but fails across all four tail transformations. This failure is characterized by a drastic suppression of the ground-truth token's probability and a flattening of the probability distribution, indicating that quantization erodes the model's predictive confidence, rendering it unable to distinguish the correct answer from competing tokens in these low-redundancy contexts.

B Diagnostic Templates and Data Construction

We construct $2,000$ diagnostic instances from 20 templates, with five templates for each of four scenarios (Table 7). Instances are generated by sampling an entity lexicon and template slots, with optional paraphrasing that preserves the query logic.

Niche varies the language of the prompt and required answer. We use Vietnamese, Thai, Japanese, Spanish, and Arabic, selected based on the observed post-quantization degradation of Qwen3-4B. **Long** introduces extended distractor passages around the relevant information. **Complex** probes less

frequent entities and relations, including time-indexed factual attributes. **Multi-hop** probes compositional queries involving intermediate entities or relations. These controlled scenarios operationalize selected aspects of the capabilities studied in Section 3; they are not intended to reproduce the full scope of the corresponding benchmarks.

We retain prediction-flipped instances, for which FP16 is correct and INT3 is incorrect. Within each scenario, 70% of instances are assigned to calibration and 30% to held-out testing. The resulting splits preserve the scenario mixture. Calibration instances are used for localization, and held-out instances for bidirectional intervention experiments. This is an instance-level split and does not establish generalization to unseen template families.

Table 7: Template examples for constructing datasets for the four tail scenarios. Slots are marked in <. . .>. **Niche** uses a <LANG> control variable (we use Vietnamese/Thai/Japanese/Spanish/Arabic because Qwen3-4B yields inferior performance on these languages after quantization). We use English templates for ease of reading, while the actual templates are in the corresponding language. **Long** inserts large distractors between the evidence and the query. **Complex** focuses on low-frequency entities/relations. **Multi-hop** targets compositional multi-hop querying (two hops or more).

Category	Templates (examples; slots in <. . .>)
Niche	[LANG=<LANG>] Fact: The capital of <COUNTRY> is <ANSWER>
	[LANG=<LANG>] Q: What is the capital of <COUNTRY>? A: <ANSWER>
	[LANG=<LANG>] Fill in the blank: Capital(<COUNTRY>) = __. Answer: <ANSWER>
	[LANG=<LANG>] Answer in <LANG>. The capital of <COUNTRY> is <ANSWER>
	[LANG=<LANG>] In <LANG>, respond with only the answer: capital of <COUNTRY> = <ANSWER>
Long	<FACT_SENTENCE>. <DISTRACTOR_LONG> Question: <QUERY>. Answer: <ANSWER>
	Context: <SHORT_FACT>. <DISTRACTOR_DOC>. Based on the context, Q: <QUERY>? A: <ANSWER>
	Key fact: <SHORT_FACT>. Read the passage. <DISTRACTOR_LONG>. Q: <QUERY>? A: <ANSWER>
	<DISTRACTOR_LONG>. Hidden note: <SHORT_FACT>. <DISTRACTOR_LONG>. Q: <QUERY>? A: <ANSWER>
	Title: <TITLE>. <DISTRACTOR_LONG>. The title is: <ANSWER>
Complex	The <ATTRIBUTE> of <COUNTRY_RARE> is <ANSWER>
	Q: What is the <ATTRIBUTE> of <COUNTRY_RARE>? A: <ANSWER>
	<COUNTRY_RARE>'s <ATTRIBUTE> is <ANSWER>
	The <ATTRIBUTE> of <COUNTRY_RARE> in <YEAR> is <ANSWER>
	<PLACE_RARE> is located in <ANSWER>
Multi-hop	Q: What is the <ATTRIBUTE> of the country containing <PLACE>? Answer directly. A: <ANSWER>
	Fact: the <ATTRIBUTE> of the <SUBJECT> that matches <CONDITION> is <ANSWER>
	Q: When did the people who <CONDITION_1> come to the <PLACE> that <CONDITION_2>? A: <ANSWER>
	<SUBJECT> is also known as <ALIAS>. What is the <ATTRIBUTE> of <ALIAS>? Answer: <ANSWER>
	Among entities related to <SUBJECT>, tell me the <ATTRIBUTE> of the one that is <CONDITION>: <ANSWER>

C Error Taxonomy under Selective Quantization

We compare four configurations: FP16, attention-only INT3 quantization (`AttnQuant`), MLP-only INT3 quantization (`MLPQuant`), and joint attention–MLP INT3 quantization (`INT3`). Other parameters remain at their reference precision. The analysis is restricted to prediction-flipped instances: FP16 is correct and joint INT3 is incorrect.

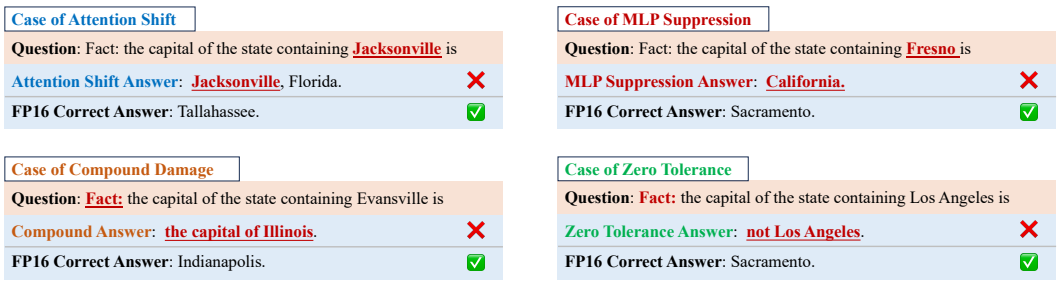


Figure 8: Representative examples of the four failure categories under selective attention and MLP quantization.

Failure categories. Let $c_v(x) \in \{0, 1\}$ indicate whether configuration v answers input x correctly. The outcomes of the two selective interventions define four mutually exclusive and exhaustive categories:

- **Attention-only:** $c_{\text{AttnQuant}}(x) = 0$ and $c_{\text{MLPQuant}}(x) = 1$.
- **MLP-only:** $c_{\text{AttnQuant}}(x) = 1$ and $c_{\text{MLPQuant}}(x) = 0$.
- **Joint-only:** $c_{\text{AttnQuant}}(x) = 1$ and $c_{\text{MLPQuant}}(x) = 1$, while joint quantization induces an error.
- **Either-alone:** $c_{\text{AttnQuant}}(x) = 0$ and $c_{\text{MLPQuant}}(x) = 0$.

These labels describe intervention outcomes, not unique internal mechanisms. In particular, joint-only failures can reflect accumulated perturbations or cross-module interactions; binary correctness alone does not distinguish them.

Illustrative cases. Figure 8 uses queries of the form "the capital of the state containing [city]." For Jacksonville, attention-only quantization produces "Jacksonville, Florida" rather than "Tallahassee." For Fresno, MLP-only quantization produces "California" rather than "Sacramento." For Evansville, both selective configurations remain correct, but joint INT3 produces an incorrect completion such as "the capital of Illinois." For Los Angeles, both selective configurations fail, with outputs such as "not Los Angeles."

The taxonomy identifies which module-type interventions are sufficient to reproduce each failure. It complements the module-level analysis without assuming that answer text alone reveals the underlying routing or transformation mechanism.

Failure modes beyond factual recall. We extend the intervention analysis to GSM8K and HumanEval, measuring recovery of the INT3–FP16 negative log-likelihood (NLL) gap on reference solutions and code. We score completion tokens under teacher forcing, using identical reference prefixes across conditions. Direction and Magnitude restore the corresponding component of selected contribution vectors, while Full restores both. Random applies full restoration to randomly selected modules matched by type and count. Both direction and magnitude repair reduce NLL, while Full recovers 60–64% of the gap, outperforming Random (Figure 9). These results support the relevance of the identified distortions beyond factual recall; the remaining gap may reflect task-specific sensitivity or effects outside the intervention scope.

D PINPOINT QUANTIZATION DETAILS

The mechanistic analysis operates on attention heads and MLP layers, with neuron-level inspection within selected MLPs. To support practical deployment, we instead use complete Linear-layer weight matrices as quantization units, avoiding intra-matrix mixed precision. Thus, the method transfers the intervention-based scoring principle rather than directly reusing the head- and neuron-level rankings. Following the weight-only quantization setting of Frantar et al. (2023), we quantize Linear-layer weights within Transformer blocks while retaining embeddings, the LM head, normalization parameters, and biases in FP16.

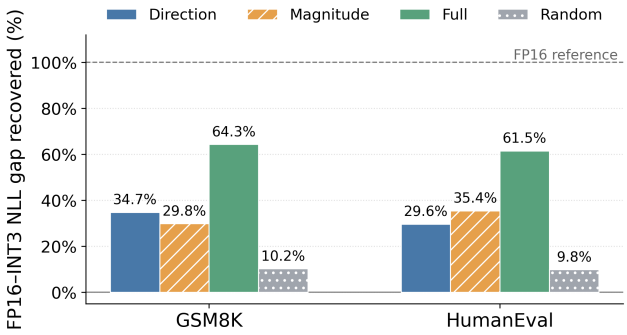


Figure 9: Gap recovery on GSM8K and HumanEval under direction, magnitude, full, and matched random repair.

Algorithm 1 Pinpoint Quantization

Require: FP16 model f, calibration set $\mathcal{D}$, allocation units $\{W_i\}_{i=1}^{N}$, candidate bit widths $\mathcal{B}$, weight-bit budget C, group size G

Initialize $s_i(b) \leftarrow 0$ for all i, b
for each input sequence $x \in \mathcal{D}$ **do**
  Compute FP16 output distributions p_t
  for each allocation unit i and $b \in \mathcal{B}$ **do**
    Construct $f_{i \leftarrow b}$ by quantizing only W_i to b bits
    Compute perturbed output distributions $p_t^{(i,b)}$
    $s_i(b) \leftarrow s_i(b) + \mathrm{Mean}_t \, D_{\mathrm{KL}}\left(p_t \| p_t^{(i,b)}\right)$
  end for
end for
$s_i(b) \leftarrow s_i(b)/|\mathcal{D}|$ for all i, b
Set $c_i(b) \leftarrow n_i b$, where $n_i = |W_i|$
Find π by dynamic programming: minimize $\sum_i s_i(\pi(i))$ subject to $\sum_i c_i(\pi(i)) \leq C$
for each allocation unit i **do**
  Quantize and pack W_i using bit width $\pi(i)$ and group size G
end for
Return quantized model $f^{(q)}$

We analyze heads and neurons for functional interpretability in Section 4.1, but allocate precision to complete weight matrices for efficient deployment. Head-level or neuron-level allocation would generally introduce intra-matrix mixed precision and additional implementation complexity. We therefore recompute causal-effect scores at the matrix level, transferring the intervention-based scoring principle rather than the fine-grained rankings.

D.1 Accelerating Intervention Scoring with LayerCache

LayerCache reduces redundant computation during intervention scoring by reusing unchanged FP16 prefix activations. For an intervention in layer k, all preceding layers are unaffected. We therefore cache the FP16 activation entering layer k and recompute only the perturbed layer and its suffix to obtain the final output distribution (Figure 10).

Computational cost. Consider a model with L layers and N intervention trials per layer. Under an equal-cost-per-layer approximation, an uncached implementation performs NL full-model evaluations, requiring

$$\mathcal{C}_{\mathrm{base}} = NL^2 \tag{5}$$

layer evaluations. With LayerCache, constructing the reference activations requires at most one FP16 traversal, while each intervention at layer k still requires evaluating layers k through L. The resulting

cost is

$$\mathcal{C}_{\text{cache}} = L + N\sum_{k=1}^{L}(L-k+1) = L + \frac{NL(L+1)}{2}. \tag{6}$$

Thus, both implementations have $\mathcal{O}(NL^2)$ computational complexity. Prefix caching provides a constant-factor reduction, with the layer-evaluation ratio

$$\frac{\mathcal{C}_{\text{base}}}{\mathcal{C}_{\text{cache}}} = \frac{NL}{1 + N(L+1)/2} \longrightarrow 2 \quad \text{as } L \to \infty \tag{7}$$

for fixed N. It does not remove the quadratic dependence on model depth, because the perturbed suffix must be recomputed for every intervention.

Cache memory. We process target layers sequentially and advance the cached reference activation using the unperturbed FP16 computation. This requires retaining only the reference activation for the current target layer rather than caching activations for all layers. For fixed batch size, sequence length, and hidden dimension, the additional persistent activation-cache memory is therefore independent of model depth. This excludes model weights and the working memory required for suffix evaluation.

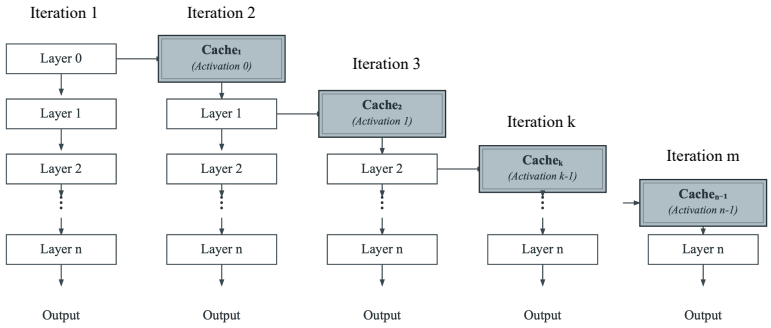


Figure 10: LayerCache reuses unchanged FP16 prefix activations during intervention scoring. For an intervention in layer k, preceding layers are bypassed, while layer k and its suffix are recomputed. This reduces redundant computation by a constant factor without changing its asymptotic dependence on model depth.

Measured runtime. On Qwen3-235B-A22B with 94 layers and 10 trials per layer, the uncached procedure requires 940 full-model evaluations. LayerCache reduces the measured analysis time from 9 hours to approximately 5 hours, yielding a $1.8\times$ end-to-end speedup. This result is consistent with the constant-factor reduction in layer evaluations derived above; the exact runtime improvement also depends on layer costs and implementation overhead.

D.2 Dynamic Programming for Bit Allocation

We formulate bit allocation as a multiple-choice knapsack problem. For allocation unit i, let $s_i(b)$ denote its intervention score at bit width b, and let $c_i(b) = n_i b$ denote its nominal weight-bit cost, where n_i is the number of weights. The objective is to minimize the summed score under budget C, selecting exactly one bit width for each unit.

Cost rescaling. To reduce the state space without changing the feasible assignments, we compute

$$g = \gcd\{c_i(b) : i = 1, \ldots, N,\ b \in \mathcal{B}\}, \tag{8}$$

and use scaled costs $\tilde{c}_i(b) = c_i(b)/g$ and capacity $\tilde{C} = \lfloor C/g \rfloor$. The capacity need not enter the GCD computation: because every attainable total cost is a multiple of g, flooring the scaled capacity preserves feasibility exactly.

DP recurrence. Let $\mathrm{DP}[i][k]$ denote the minimum summed score achievable using the first i units with scaled total cost exactly k. Initialize $\mathrm{DP}[0][0] = 0$ and $\mathrm{DP}[0][k] = +\infty$ for $k > 0$. The recurrence is

$$\mathrm{DP}[i][k] = \min_{\substack{b \in \mathcal{B} \\ \tilde{c}_i(b) \le k}} \{\mathrm{DP}[i-1][k - \tilde{c}_i(b)] + s_i(b)\}. \tag{9}$$

The best feasible objective is

$$\min_{0 \le k \le \tilde{C}} \mathrm{DP}[N][k], \tag{10}$$

and the assignment is recovered by backtracking. This procedure is exact for the additive scoring objective, rather than for downstream accuracy, since joint quantization effects may be non-additive.

Computational requirements. The dense DP requires $\mathcal{O}(N|\mathcal{B}|\tilde{C})$ time and $\mathcal{O}(N\tilde{C})$ memory when the full table is retained for backtracking. Its practicality therefore depends on the scaled capacity, not solely on the number of allocation units. GCD rescaling is an exact compression of the cost axis, but not a general memory guarantee: irregular parameter counts can yield a small GCD and leave a prohibitively large state space. The dense formulation is consequently applicable when the rescaled table fits the available memory; models outside this regime require a different solver or an explicitly approximate cost discretization.

D.3 Inference Kernels

We execute quantized matrix multiplications using Marlin GEMM kernels (Frantar et al., 2025) integrated with vLLM. We use standard kernels for 4- and 8-bit weights and custom variants for 2-, 3-, and 6-bit weights. Dequantization is fused into GEMM, with group-wise scales and zero-points stored alongside packed weights.

The custom variants retain Marlin's pipelined memory staging while adapting tiling, addressing, and bit extraction to the supported formats. For 3-bit weights, each 16-bit fragment stores five values, giving ten values per 32-bit word. For 6-bit weights, each 32-bit word stores five values. Both formats therefore store 30 payload bits per 32 bits.

Padding occupies 6.25% of the allocated words, equivalent to a 6.67% storage increase over dense bit packing. The corresponding packed weight costs are 3.2 and 6.4 bits per weight, before scales, zero-points, and other metadata.

D.4 Group-Wise MinMax Quantization

We use group-wise affine MinMax quantization. For a weight matrix $W_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we partition each row into groups of G consecutive weights along the input dimension. For assigned bit width $b = \pi(i)$, the integer range is $q_{\min} = 0$ to $q_{\max} = 2^b - 1$.

For each group $W_{i,g}$, the quantization bounds are determined by its minimum and maximum values:

$$\ell_{i,g} = \min(W_{i,g}), \qquad u_{i,g} = \max(W_{i,g}). \tag{11}$$

The group-wise scale and zero-point are

$$s_{i,g} = \max\left(\frac{u_{i,g} - \ell_{i,g}}{q_{\max} - q_{\min}}, \epsilon\right), \qquad z_{i,g} = \mathrm{clip}\left(\mathrm{round}\left(q_{\min} - \frac{\ell_{i,g}}{s_{i,g}}\right), q_{\min}, q_{\max}\right), \tag{12}$$

where $\epsilon > 0$ is a small constant for numerical stability.

Quantization and reconstruction are performed as

$$Q_{i,g} = \mathrm{clip}\left(\mathrm{round}\left(\frac{W_{i,g}}{s_{i,g}}\right) + z_{i,g}, q_{\min}, q_{\max}\right), \tag{13}$$

$$\widehat{W}_{i,g} = s_{i,g}(Q_{i,g} - z_{i,g}). \tag{14}$$

The integer weights are packed at the assigned precision and stored with group-wise scales and zero-points. During inference, dequantization is fused into the matrix multiplication.

E Implementation Details

We quantize Linear-layer weights within Transformer blocks, retaining embeddings, the LM head, normalization parameters, and biases in FP16. The default quantizer is group-wise MinMax with zero-points and group size $G = 128$. Candidate bit widths are $\mathcal{B} = \{2, 3, 4, 6, 8\}$.

For the main Qwen3 evaluation, the target average precisions are 3.3, 3.5, and 3.8 bits for Qwen3-235B-A22B, Qwen3-30B-A3B, and Qwen3-4B, respectively. Other budgets are specified in the corresponding comparison and ablation tables.

The allocation objective uses the nominal weight-bit average

$$\bar{b} = \frac{\sum_i n_i \pi(i)}{\sum_i n_i}, \tag{15}$$

over quantized allocation units. This quantity excludes packing padding, quantization metadata, and parameters retained in FP16. Actual memory measurements are reported separately.

Calibration data. Allocation scores are computed using 2,048 sequences of 512 tokens, formed by concatenating and truncating text from the 11 sources in Table 8. This mixture is distinct from the prediction-flipped diagnostic dataset used for mechanistic localization and intervention-based validation.

Fused execution. Following the vLLM execution structure (Kwon et al., 2023), we fuse the $\{W_Q, W_K, W_V\}$ projections and the $\{W_{\text{gate}}, W_{\text{up}}\}$ projections. For MoE models, every 32 experts form a group, within which gate/up and down projections are fused separately. These implementation choices reduce the number of separately executed linear operations.

Table 8: Calibration data mixture for PinQ allocation. All 2,048 sequences contain 512 tokens.

Dataset	#Sequences	Proportion (%)
Wikitext-2 Merity et al. (2017)	128	6.25
C4 (Raffel et al., 2020)	128	6.25
mC4 (Xue et al., 2021)	256	12.50
Alpaca (Taori et al., 2023)	256	12.50
APIGen (Liu et al., 2024b)	256	12.50
hh-rlhf (Bai et al., 2022)	256	12.50
OpenWebMath (Paster et al., 2024)	128	6.25
S1k-v1.1 (Muennighoff et al., 2025)	128	6.25
StarCoder (C++) (Li et al., 2023)	128	6.25
StarCoder (Python) (Li et al., 2023)	128	6.25
UltraChat (Ding et al., 2023)	256	12.50
Total	2,048	100.00

F Detailed Results

F.1 Bit Assignment Results

Figures 11 and 12 show the bit allocation for Qwen3-235B-A22B-Instruct-2507. The assignment exhibits a broadly U-shaped depth profile: early and late layers generally receive higher precision, while many middle-layer expert groups receive 2 or 3 bits. Attention projections predominantly receive 6 or 8 bits. This depth profile is therefore not uniform across module types: low-bit middle-layer experts coexist with higher-precision attention projections.

Relation to mechanistic localization. This allocation should not be read as a direct translation of the Qwen3-4B localization maps. The mechanistic analysis measures fixed-precision intervention effects on prediction-flipped examples from four tail scenarios, primarily at head and MLP-layer granularity. Deployment allocation instead recomputes intervention scores for Qwen3-235B-A22B at the allocation-unit level, across multiple candidate precisions and a mixed calibration corpus. PinQ therefore transfers the intervention-based scoring principle, not a fixed layer ranking.

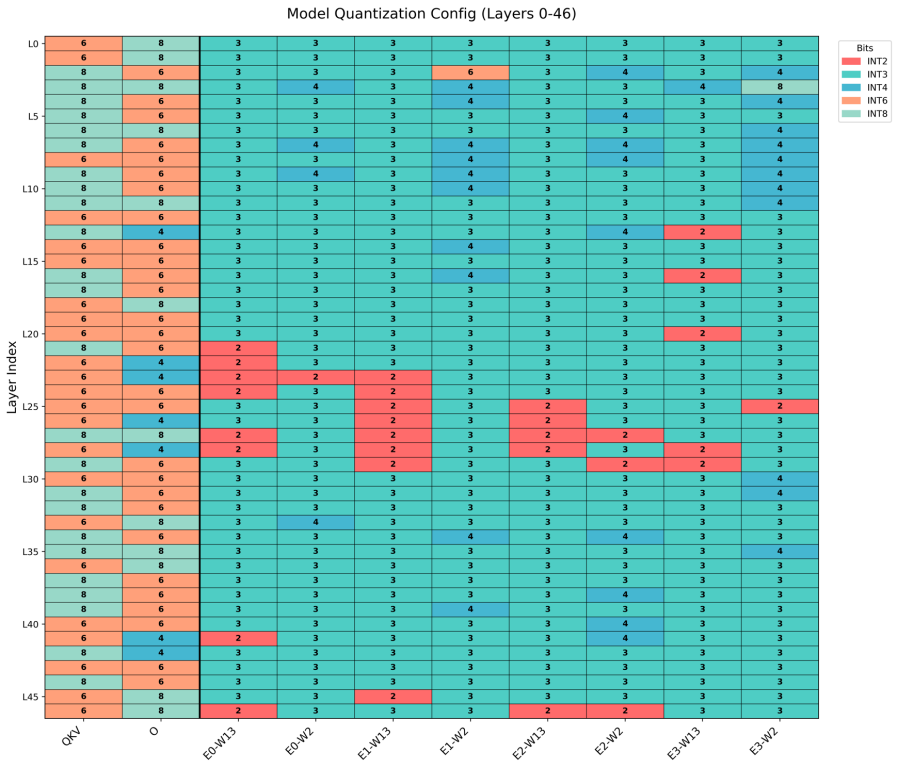


Figure 11: Bit-width allocation for layers 0–46 of Qwen3-235B-A22B. Attention projections predominantly receive 6 or 8 bits, while many middle-layer expert groups receive 2 or 3 bits.

Budget-dependent precision trade-offs. The DP planner minimizes the summed intervention scores subject to a global weight-bit budget. Its decisions depend on both the score reduction obtained by increasing precision and the associated storage cost. For a unit with n_i weights, moving from b to b' bits costs $n_i(b' - b)$ additional bits. Consequently, neither a large effect at one precision nor a low assigned bit width alone determines a unit's importance. In particular, the low-bit assignments of middle-layer expert groups represent budget-constrained choices, not evidence that these groups are functionally redundant or individually insensitive to quantization.

Implications for tail-capability preservation. Because the deployment objective averages distributional effects over mixed calibration data, it does not explicitly guarantee preservation of every tail capability. The Qwen3-235B-A22B results show that the resulting 3.3-bit allocation retains aggregate performance close to FP16, but residual degradation remains: LongBenchV2 decreases from 35.0 to 33.0 (Table 3). Thus, the allocation provides a practical balance between compression and overall quality, rather than uniformly lossless capability preservation. These results evaluate the complete allocation; they do not isolate the contribution of its U-shaped depth profile to any individual capability.

F.2 Inference Efficiency

We evaluate Qwen3-235B-A22B under FP16 and PinQ at a nominal average precision of 3.3 bits on 8 NVIDIA A100 80GB GPUs. Both configurations use vLLM (Kwon et al., 2023), with custom Marlin kernels for PinQ. The serving benchmark uses a prompt length of 512 tokens, an output length of 128 tokens, and concurrency levels from 2^2 to 2^7.

Table 9 reports memory usage and serving performance. At concurrency 128, PinQ increases token throughput from 1,149 to 1,472.4 tokens/s (1.28×), reduces time to first token (TTFT) from 3,413.5 to 1,625.5 ms, and reduces time per output token (TPOT) from 83.5 to 74.68 ms. Reported memory

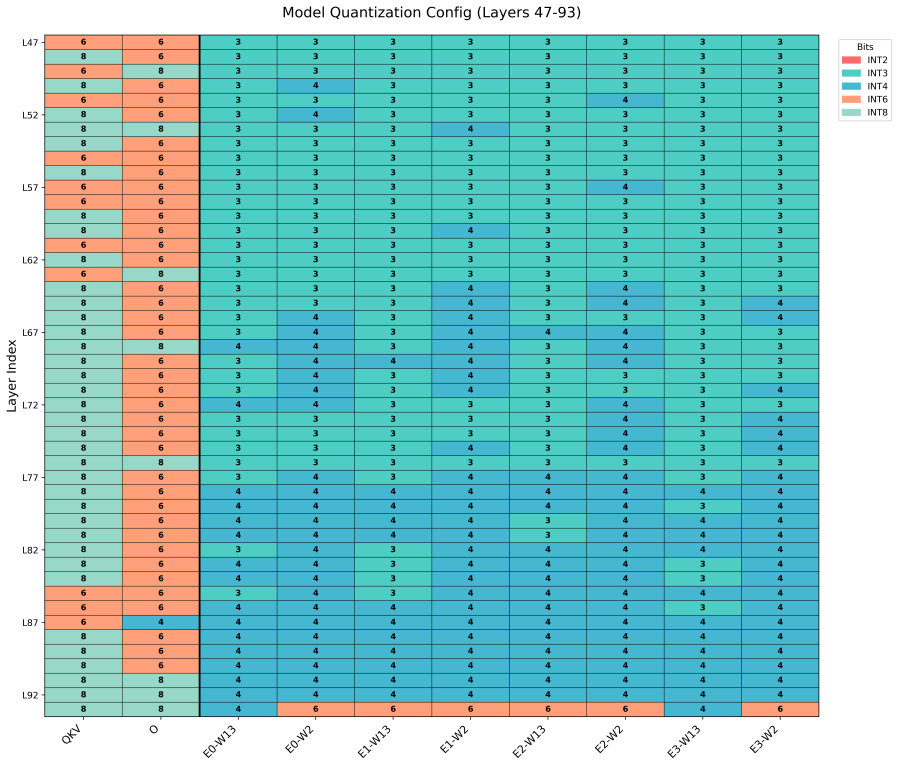
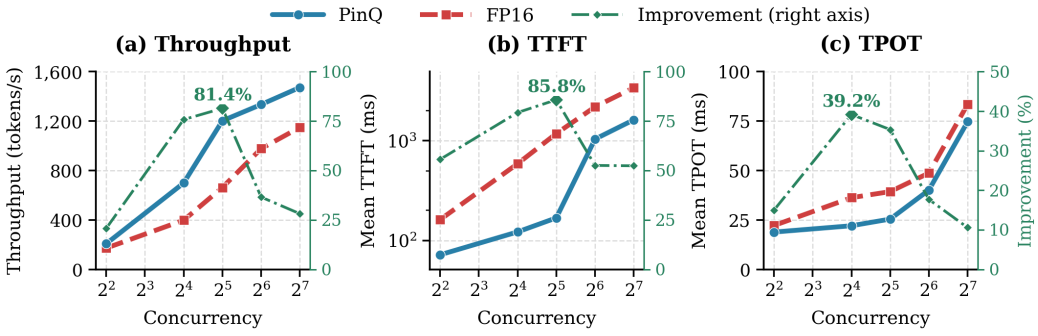


Figure 12: Bit-width allocation for layers 47–93 of Qwen3-235B-A22B. Expert-group precision generally increases in the later layers, with many groups assigned 6 bits in the final layer. Attention projections continue to receive predominantly 6 or 8 bits.

Table 9: Inference efficiency of Qwen3-235B-A22B with FP16 and PinQ at 3.3 average bits on 8 NVIDIA A100 80GB GPUs. Top: reported memory usage and serving metrics at concurrency 128. Bottom: serving performance across concurrency levels from 2^2 to 2^7.

Configuration	Memory (GB) ↓	Throughput (tokens/s) ↑	TTFT (ms) ↓	TPOT (ms) ↓
FP16	437.90	1149	3413.5	83.5
PinQ (3.3 bits)	96.78	1472.4	1625.5	74.68



usage decreases from 437.90 to 96.78 GB, a reduction of approximately 78%. These measurements show that the mixed-precision configuration provides serving benefits in addition to preserving downstream accuracy.