Fractional Power-of-Two Quantization for Efficient and Effective Multiplier-Free LLM Inference

Sunghyun Wee
Seoul National University
LG Electronics
wsh05@snu.ac.kr

Geunjae Choi*
LG Electronics
cgj.choi@lge.com

Hyeonjin Kim*
Seoul National University
peaceful1@snu.ac.kr

Suyoung Kim*
Seoul National University
ksyo96@snu.ac.kr

Nojun Kwak†
Seoul National University
nojunk@snu.ac.kr

Abstract

Large Language Models (LLMs) demand efficient inference, where the multiply-accumulate (MAC) operations in linear projections dominate compute and energy cost. Post-training quantization (PTQ) reduces this cost by mapping weights and activations to low-bit integers. Power-of-Two (PoT) weight quantization promises efficient multiplier-free LLM inference via shift-add logic, but suffers significant accuracy degradation at low bit-widths due to its coarse base-2 quantization grid. We introduce **Fractional PoT (FPoT)**, a base-$\sqrt{2}$ (half-octave) fractional PoT grid that achieves practical 4-bit weight quantization while preserving a fully integer shift-add datapath. The method combines (i) the base-$\sqrt{2}$ weight grid with uniform integer activations, (ii) a calibration-time grid-alignment refinement absorbed into the static per-channel scale, and (iii) a lightweight 1-adder $\sqrt{2}$ approximation whose error is absorbed during weight calibration via approximate-grid quantization, yielding a processing element (PE) with significantly fewer gates than a 4-bit integer multiplier. On Llama-3-8B with 4-bit weight and activation quantization, FPoT improves zero-shot commonsense reasoning performance by 8.72% relative to the PoT baseline, while maintaining performance close to the uniform integer baseline, with only a negligible 0.4%p degradation. These results hold across various configurations spanning seven models and three bit-widths. Furthermore, FPoT can be integrated with state-of-the-art LLM PTQ methods, demonstrating framework-agnostic applicability.

1 Introduction

The rapid proliferation of Large Language Models (LLMs) [Touvron et al., 2023, Grattafiori et al., 2024, Yang et al., 2024, Jiang et al., 2023] has transformed natural language processing, yet their deployment on resource-constrained edge devices remains challenging. One of the dominant costs in LLM inference is the multiply-accumulate (MAC) operation within Transformer [Vaswani et al., 2017] linear projections, which account for virtually all parameters and a majority of inference energy [Wang et al., 2020, Sze et al., 2017]. Recent post-training quantization (PTQ) [Ashkboos et al., 2024, Liu et al., 2025, Hu et al., 2025, Lin et al., 2024a] has pushed integer (INT) quantization to 4-bit weights, activations, and key-value (KV) cache at near-lossless accuracy, making 4-bit the practical operating point: 8-bit pays with costly multipliers, while lower bits collapse LLM accuracy.

*Equal contribution
†Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Power-of-Two (PoT) quantization, also known as logarithmic quantization [Elhoushi et al., 2021, Li et al., 2020], offers a multiplier-free pathway and further reduces computation: constraining weights to $\pm 2^n$ allows MAC operations to be substituted with bit-wise shifts and an integer addition, substantially reducing the per-MAC energy. Register-Transfer Level (RTL) synthesis on Samsung 5 nm shows a PoT weight MAC at $0.42\times$ the energy of an equivalent INT MAC [Przewlocka-Rus et al., 2022].

Though efficient, accuracy remains the central obstacle of PoT quantization. Standard PoT places representable levels at geometric intervals of factor 2 (one octave), yielding a worst-case rounding error of 41.4% ($2^{1/2}-1$) at the midpoint under a 4-bit round-to-nearest strategy. While this coarseness is tolerable for Convolutional Neural Networks (CNNs) [Elhoushi et al., 2021, Li et al., 2020, 2023], it causes catastrophic degradation in LLMs: perplexity (PPL) degradation of $+9.5$ to $+107.5$ when PoT is applied to both weights and activations, or $+1.43$ when restricted to weights alone. *Standard PoT is not a viable drop-in replacement for multiplication operations in LLMs.*

In this work, we show that the accuracy barriers of 4-bit PoT stem from a grid mismatch with the LLM weight statistics, and introduce **Fractional PoT (FPoT)** to address this. On Llama-3-8B at 4-bit, applying FPoT to weights reaches on-par performance with the uniform INT baseline (only 0.4%p 0-shot degradation), while delivering an 8.72% relative gain over standard PoT. Throughout this paper, we use *near-lossless* to denote configurations whose perplexity increase remains within $+0.5$ and whose downstream accuracy drop remains within 1.0%p of the uniform INT baseline.

Our contributions are as follows.

- **Theoretically grounded base-$\sqrt{2}$ grid for 4-bit LLMs**: We conduct a Gaussian-proxy analysis to identify base-$\sqrt{2}$ as the optimal integer-resolution PoT grid at 4-bit, halving the worst-case relative error of standard PoT (41.4%→18.9%).
- **Approximate-grid quantization with low computational overhead**: FPoT quantizes weights to the actual hardware grid using a 1-adder $\sqrt{2}$ approximation. The resulting processing element (PE) requires significantly fewer gates than a 4-bit integer multiplier.
- **Near-lossless performance over uniform INT baselines**: FPoT integrates with state-of-the-art LLM PTQ frameworks at comparable overhead (Appendix J), achieving performance on par with uniform INT baselines.

2 Related Work

2.1 Post-Training Quantization (PTQ) for LLMs

Uniform INT weight quantization methods such as GPTQ [Frantar et al., 2023] and AWQ [Lin et al., 2024b] achieve competitive accuracy at 4-bit via Hessian-weighted error compensation and activation-aware weight scaling. Based on these works, a line of research on quantizing activations to low bit-widths emerged for more efficient calculation. However, activation quantization is more challenging due to the high-magnitude outliers concentrated in a small subset of channels [Dettmers et al., 2022]. Extreme outliers cause uniform INT4 activation quantization to collapse most activations to zero under a per-tensor scale dominated by the outlier channels. Gradually, attempts to mitigate such outliers emerged. Early approaches such as SmoothQuant [Xiao et al., 2023] migrate the quantization difficulty from activations to weights via per-channel scaling. Recent methods resolve this via rotation-based preprocessing that redistributes outliers across dimensions: QuaRot [Ashkboos et al., 2024] applies randomized Hadamard rotations, SpinQuant [Liu et al., 2025] learns globally optimal rotations, OSTQuant [Hu et al., 2025] jointly trains rotation with per-channel smooth scales for tighter control, and DuQuant [Lin et al., 2024a] adds dual block-diagonal transforms and channel permutation. These approaches collectively enable 4-bit weights, 4-bit activations, 4-bit KV cache (W4A4KV4) uniform quantization at near-lossless accuracy.

While the aforementioned methods enable practical 4-bit activation quantization, they mainly focus on uniform quantization grids. In this work, we apply logarithmic weight quantization to the state-of-the-art quantization methods and demonstrate that it can reach comparable performance with better efficiency.

2.2 Power-of-Two (PoT) and Shift-based Quantization

PoT quantization [Elhoushi et al., 2021] is a family of methods that constrains weights to $\pm 2^n$ values and replaces multipliers with barrel shifters. Prior works on CNN include DeepShift [Elhoushi et al., 2021], APoT [Li et al., 2020], and DenseShift [Li et al., 2023]. These methods are either not directly applicable to LLMs, or represent weights as sums of multiple PoT terms, requiring n sequential shift-adds per MAC. For LLMs, PoT-PTQ [Wang et al., 2025] is a base-2 quantization that achieves 2–3-bit via data-agnostic scale initialization plus residual fine-tuning. Concurrent with our work, LogART [Xu et al., 2026] extends logarithmic PTQ with learnable rounding and dynamic base allocation between base-2 and base-$\sqrt{2}$, realizing $\sqrt{2}$ with the shift-add approximation $\sqrt{2} \approx 2^0 + 2^{-1}$ of earlier logarithmic accelerators [Xu et al., 2018, 2023]. However, both methods validate their LLM accuracy only with floating-point (FP) activations at 16-bit precision (FP16).

In that setting, the main operation is `FP16 × PoT`, requiring FP16 accumulators that cannot be simplified to an integer shift-add unit. Our FPoT pairs a fixed half-octave grid with uniform INT activations, enabling a fully integer shift-add PE.

Position of this work. Between the two distinct lines of work—activation quantization, which enables fully INT-only arithmetic, and PoT quantization, which replaces multipliers with shift-add PE—a gap remains. Our method, FPoT, aims to address this gap: a multiplier-free LLM quantization method that retains uniform INT activations and composes with recent PTQ frameworks. State-of-the-art uniform INT quantization methods [Ashkboos et al., 2024, Liu et al., 2025, Hu et al., 2025, Lin et al., 2024a] achieve near-lossless W4A4 but retain integer multipliers. PoT-based methods are either CNN-targeted [Elhoushi et al., 2021, Li et al., 2020, 2023], or validate LLM accuracy only with FP activations [Wang et al., 2025, Xu et al., 2026]. FPoT can pair base-$\sqrt{2}$ weight quantization with state-of-the-art uniform INT activation quantization to achieve efficient and effective W4A4 quantization. $\sqrt{2}$ is approximated with a single adder, where the approximation cost is absorbed via approximate-grid quantization (§3). FPoT realizes a fully integer shift-add PE at near-lossless W4A4KV4 accuracy.

3 Methodology

Our method applies base-$\sqrt{2}$ quantization to all linear projection weights of each Transformer block while maintaining uniform INT quantization for activations and the key-value (KV) cache. This design isolates the logarithmic grid to static weights, where calibration can absorb quantization error, while preserving hardware-friendly uniform INT formats for dynamic activations and KV cache. Weight distributions are redistributed by state-of-the-art LLM PTQ preprocessing (Appendix A) prior to GPTQ calibration, and a calibration-time weight grid-alignment further refines the per-channel scale. In brief, such methods apply learned rotations and per-channel scaling to redistribute channel-concentrated outliers, producing smoother distributions amenable to the logarithmic grid. The transform adds negligible overhead ($\approx 0.02\%$ of weight MACs).

Grid resolution and formulation. Standard PoT quantization places representable levels at geometric intervals of factor 2 (one octave), yielding a worst-case relative quantization error of $2^{1/2} - 1 \approx 41.4\%$ at the midpoint between adjacent levels. This coarse resolution is a fundamental accuracy bottleneck for LLM weight quantization using PoT.

We introduce a *fractional exponent resolution parameter* $k \in \mathbb{Z}_{>0}$ (FPoT), an extension of the PoT grid. For a per-channel scale $s > 0$ and n levels per sign, the FPoT quantized value is:

$$Q_{\mathrm{FPoT}}(w;\, s, n, k) = \mathrm{sgn}(w) \cdot s \cdot 2^{\,\mathrm{clip}\left(\left\lfloor k \cdot \log_2 \frac{|w|}{s} \right\rceil,\, 0,\, n-1\right)/k}, \tag{1}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer, and the exponent index is clipped to $[0,\, n{-}1]$ before division by k. For $k{=}1$, Eq. (1) reduces to standard PoT. For $k{=}2$ (half-octave grid), representable values are spaced at intervals of $2^{1/2} \approx 1.414$, reducing the worst-case relative error to $2^{1/4} - 1 \approx 18.9\%$—roughly a $2.2\times$ improvement in grid resolution. We analyze the optimality of $k{=}2$ for 4-bit LLM weight quantization in §4.1.

Signed quantization with k-adaptive zero threshold. The zero threshold follows naturally from the grid geometry. For a FPoT grid with minimum exponent index $e{=}0$, the log-domain decision

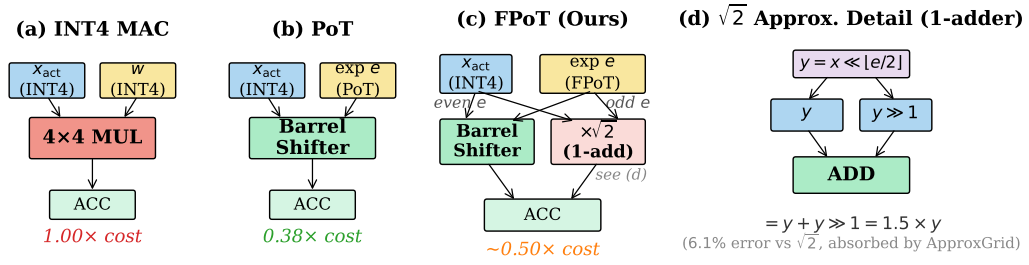


Figure 1: PE architecture: INT4 multiplier, PoT barrel shifter, and FPoT (Ours). FPoT adds one adder and a MUX; $\sqrt{2}\,y \approx y + y{\gg}1 = 1.5y$ (6.1% error, absorbed by approximate-grid quantization, §3).

boundary between "round to e=0" and "round to zero" yields:

$$\tau_k = 2^{-1/(2k)}, \tag{2}$$

so that $|w|/s < \tau_k$ maps to zero. At k=2, $\tau_2 \approx 0.841$; the resulting fraction of weights mapped to zero and its accuracy implications are analyzed in §4.1.

The complete FPoT quantization procedure, including signed representation, k-adaptive zero thresholding, and the inference-time hardware decomposition, is given in the pseudocode in Appendix F.

Hardware implications. The stored weight is encoded as b bits with one sign bit and a magnitude code $m \in \{0, \ldots, 2^{b-1}{-}1\}$. When m=0 the MAC is skipped (zero-gating); otherwise the exponent index $e = m{-}1$ yields the dequantized value $\hat{w} = \mathrm{sgn}(w) \cdot s \cdot 2^{e/k}$. For k=2, $\lfloor e/2 \rfloor$ and $e \bmod 2$ are direct wire selections from the magnitude bits (Appendix F). Combining FPoT weights with uniform INT activations, the shift-accumulate MAC becomes:

$$y = \sum_i \mathrm{sgn}(w_i) \cdot (x_i \cdot 2^{\lfloor e_i/2 \rfloor}) \cdot (\sqrt{2})^{e_i \bmod 2}, \tag{3}$$

which replaces integer multipliers with barrel shifters plus a lightweight $\times\sqrt{2}$ unit for odd exponents ($\sim$50% of MACs). Activations remain in uniform INT format (Appendix D provides the supporting per-projection analysis). The entire FPoT compute unit shares a **single barrel shifter** with PoT; the only additional logic is a 1-adder $\sqrt{2}$ approximation unit and a multiplexer (MUX) selected by the least significant bit (LSB) of the weight (Fig. 1). This shift-accumulate structure replaces all 4-bit integer multipliers in every linear projection of each Transformer block with a uniform FPoT PE.

Approximate-grid quantization. The odd-e path above requires a $\times\sqrt{2}$ coefficient. A 1-adder realization ($x + x{\gg}1 = 1.5x$) deviates from $\sqrt{2} \approx 1.414$ by 6.1%, which would normally degrade accuracy under naïve post-hoc substitution—quantizing to the exact FPoT grid and only replacing the coefficient at dequantization. Our solution is *approximate-grid quantization*: we calibrate weights on the *actual hardware grid* (i.e., using 1.5 for odd e instead of $\sqrt{2}$), so that weight calibration absorbs the approximation as part of the quantization grid rather than as post-hoc distortion. Empirical validation of this scheme is reported in §5.3, with the resulting hardware cost analyzed in §4.2.

Weight grid-alignment. We additionally evaluate a calibration-time grid-alignment refinement applicable to both PoT and FPoT grids. Since weights are static after calibration, the optimal phase offset can be absorbed into the per-channel scale at **zero inference overhead**. We add a phase-shift refinement evaluating G candidate scales

$$s_g \;=\; s_{\text{base}} \cdot 2^{g/G}, \quad g = 0,\, 1,\, \ldots,\, G{-}1, \tag{4}$$

and pick per output-channel the g^* minimizing MSE between raw weights and their FPoT reconstruction. The winning s_{g^*} replaces s_{base} as a static per-channel weight scale, so the deployed architecture is identical to unaligned FPoT.

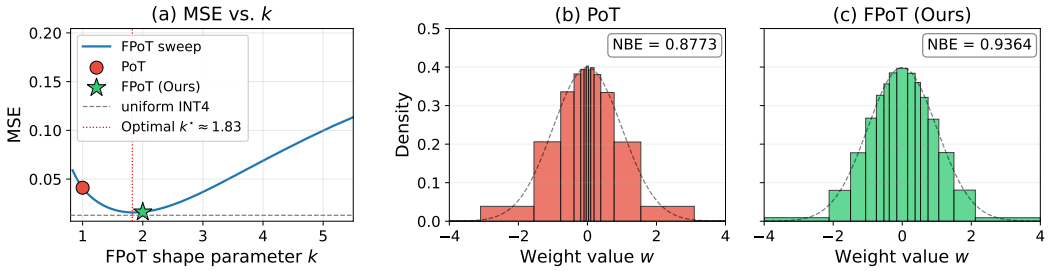


Figure 2: (a) MSE vs. shape parameter k for $\mathcal{N}(0,1)$ weights at 4-bit (N=15 levels): the MSE is minimized at $k^\star \approx 1.8$, so the integer choice k=2 is near-optimal. (b, c) Bin occupancy under the Gaussian proxy. FPoT concentrates levels near the high-density mode, while PoT spreads levels across an excessively wide range, leaving most of them in the sparsely populated tail. Dashed curves: input density p_W.

4 Analysis

We analyze why the FPoT base-$\sqrt{2}$ grid is optimal for 4-bit LLM weight quantization (§4.1) and quantify its hardware cost relative to INT and standard PoT baselines (§4.2).

4.1 Grid-Matching Analysis

Gaussian proxy. Raw LLM weights are far from Gaussian: they exhibit heavy-tailed, leptokurtic distributions with high-magnitude outliers concentrated in a few channels. Recent LLM PTQ preprocessing *redistributes* channel-concentrated outliers across dimensions, reducing kurtosis and equalizing per-channel dynamic range. We analyze a zero-mean symmetric distribution whose log-magnitude is unimodal—a property that holds for the preprocessed weights, with the Gaussian $w \sim \mathcal{N}(0, \sigma^2)$ serving as a tractable analytical proxy. Under this proxy, $u = \log_2 |w|$ is approximately Gaussian in log-space with standard deviation $1/(\sqrt{2}\ln 2) \approx 1.02$ octaves (derivation in Appendix B.2), so $\sim$95% of probability mass concentrates within $\sim$4 octaves around the mode. Empirically, the preprocessed weight distributions of all seven models studied closely match this Gaussian-proxy prediction (Appendix C), validating its use for 4-bit grid design.

Optimal base under the Gaussian proxy. A PoT quantizer with step ratio $\beta = 2^{1/k}$ and n levels per sign covers a dynamic range β^{n-1}; with n=7, k=1 yields a 64× range that exceeds the $\sim$4-octave preprocessed weight spread, k=2 yields 8× (aligned), and k=4 yields 2.8× (tail clipping). Numerically minimizing the MSE on $w \sim \mathcal{N}(0,1)$ with 15 levels while jointly optimizing the scale s yields $\beta^\star \approx 2^{0.55}$ (effective $k^\star \approx 1.8$), very close to k=2 (Fig. 2): FPoT$_{k=2}$ achieves MSE = 1.27× INT4, while PoT wastes levels near zero (3.21×) and $k \geq 3$ suffers dynamic range starvation (2.87× at k=3, 5.37× at k=4). This optimality is bit-width specific—at 8-bit the optimum shifts to $k^\star \approx 19$—so the k=2 choice is specifically motivated by the 4-bit constraint (8-bit analysis and full numerical table in Appendix B.3).

Normalized Bin Entropy. To quantify how effectively a grid utilizes its fixed bit budget under a given weight distribution, we introduce the *Normalized Bin Entropy* (NBE). For a quantizer Q with N representable levels and level-occupancy probabilities $p_i = \Pr[Q(W) = \ell_i]$, we define

$$\mathrm{NBE}(Q; p_W) \;=\; \frac{H(\mathbf{p})}{\log_2 N}, \quad H(\mathbf{p}) = -\textstyle\sum_{i=1}^{N} p_i \log_2 p_i, \tag{5}$$

where N is the budget-determined number of representable levels (e.g., N=15 for the 4-bit symmetric grids studied here: 7 positive + 7 negative + 1 zero). NBE = 1 iff data mass is uniform across all levels; NBE $\to$ 0 as mass concentrates on a single level. Information-theoretically, $\log_2 N \cdot \mathrm{NBE}$ is the expected Shannon code length per weight, so NBE measures the fraction of the fixed bit budget that carries information (Appendix B.1). Unlike volumetric metrics such as QSUR [Hu et al., 2025], NBE distinguishes grid structures sharing the same dynamic range (Property 4 in Appendix B.1).

Under the Gaussian proxy, $\text{FPoT}_{k=2}$ achieves $\text{NBE} = 0.9364$, surpassing both uniform INT4 (0.9026) and PoT (0.8773): the half-octave log-grid places denser levels where probability mass concentrates near the mode $|w| \approx \sigma$, while PoT wastes levels in the sparsely populated tail ($64{:}1$ range); Fig. 2 visualizes this contrast. Empirical bin occupancy on real LLM weights (Appendix C) closely matches the Gaussian-proxy prediction.

Implicit magnitude pruning. The k-adaptive zero threshold $\tau_k = 2^{-1/(2k)}$ (Eq. (2)) interacts with the Gaussian-proxy distribution to map a k-dependent fraction of weights to zero. At $k{=}2$, $\tau_2 \approx 0.841$ causes $\sim 22\%$ of preprocessed weights to be zeroed—**implicit magnitude pruning** that is fully compensated by the $2\times$ finer grid resolution. Empirical measurement on real LLM weights confirms this ratio (Appendix C). These zeroed weights further skip their MACs entirely via zero-gating (§3), providing additional compute reduction beyond the shift-add substitution. Prior PoT methods (DeepShift, APoT) adopt ad-hoc zero strategies without recognizing this k-dependent effect.

4.2 Hardware Cost Analysis

We evaluate the hardware benefits of FPoT weight quantization with two complementary metrics: compute-unit (CU) cost from a naïve component model and energy from published RTL synthesis data. FPoT is applied to all linear projection layers, converting all weight MACs (6.98 B per token) into shift-accumulate operations (per-projection breakdown in Appendix D).

Compute unit cost. We model hardware cost with a naïve component-based analysis (Appendix E.1): an INT multiplier grows quadratically ($W \times A$ gate-equivalents), while a PoT barrel shifter and FPoT shifter + adder grow linearly. FPoT adds one adder for the $\sqrt{2}$ approximation ($x{+}x{\gg}1 = 1.5x$, 6.1% error absorbed by approximate-grid quantization, §3). At W4A4, PoT costs $0.38\times$ and FPoT costs $\sim 0.50\times$ the INT multiplier; the $O(WA)$-vs.-$O(W{+}A)$ scaling gap widens at higher bit-widths (Table 6). Critically, the additional 1-adder over standard PoT costs only $\sim 0.12\times$ extra CU at W4A4 while reducing perplexity overhead by 67%–94%, placing FPoT at the knee of the accuracy–efficiency Pareto frontier (Fig. 3) in the compute-bound prefill regime. Open-source logic synthesis (Yosys [Wolf and Glaser, 2013]) is consistent with the structural ordering INT > FPoT > PoT and the bit-width scaling predicted above (Tables 6–7, Appendix E.2).

Energy. We anchor the energy comparison to RTL synthesis by Przewlocka-Rus et al. [2022] on Samsung 5 nm (250 MHz, TT corner, 0.6 V, 25°C, 50% toggle rate), which reports a PoT-to-INT energy ratio of $\mathbf{0.42}\times$ at W4A8—a $2.4\times$ reduction per weight MAC. FPoT with the 1-adder $\sqrt{2}$ unit adds minimal overhead (one adder, active for $\sim 50\%$ of MACs); we report the FPoT energy ratio as $\mathbf{0.42}$–$\mathbf{0.46}\times$ at W4A4 (0.84–$1.00\times$ at W4A8; 1.68–$2.00\times$ at W8A8)—a 2.2–$2.4\times$ energy reduction per weight multiply versus INT, with the $\sqrt{2}$ unit contributing at most $0.04\times$ overhead for a 50% duty cycle on odd exponents. Appendix E.4 provides per-model accuracy–cost trade-off curves.

Scope of hardware claims. The 0.42–$0.46\times$ energy and $0.50\times$ CU ratios are *per weight MAC*, anchored to cell-level RTL synthesis [Przewlocka-Rus et al., 2022]; end-to-end inference energy reduction scales with the fraction of total energy attributable to weight MACs, which is largest in the compute-bound prefill regime and diminishes under memory-bound decode (Limitations, §6).

5 Experimental Results

5.1 Experimental Setup

FPoT is framework-agnostic with respect to the choice of state-of-the-art LLM PTQ preprocessing; for the main results we use the OSTQuant framework [Hu et al., 2025] for preprocessing, with rotation parameters jointly optimized with smooth scaling factors via Riemannian gradient descent for 400 steps with straight-through estimator (STE) gradients through discrete PoT quantization. Final weights are obtained via GPTQ [Frantar et al., 2023] with PoT targets. Cross-framework portability with DuQuant is validated in Appendix J. We evaluate WikiText-2 [Merity et al., 2017] perplexity (PPL↓) and the unweighted mean of zero-shot accuracy across nine benchmarks (0-shot[9]↑, in %): ARC-Challenge, ARC-Easy [Clark et al., 2018], HellaSwag [Zellers et al., 2019], WinoGrande [Sakaguchi

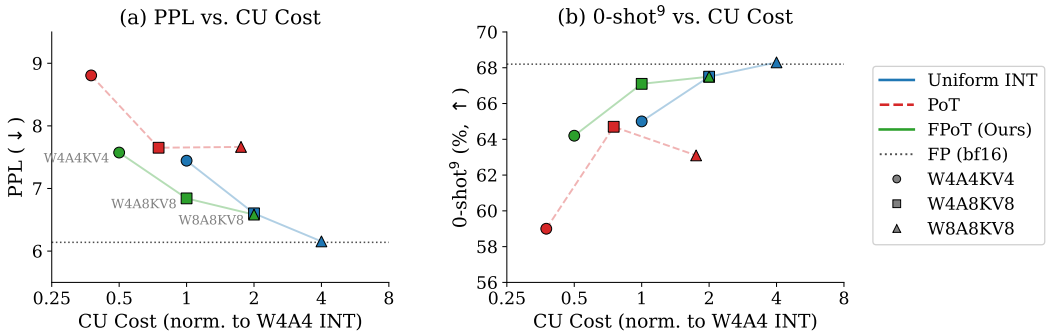


Figure 3: Accuracy–cost Pareto trade-off for Llama-3-8B. FPoT achieves near-baseline PPL at $\sim 0.50\times$ CU cost across all bit-widths. Per-model plots in Appendix E.4.

et al., 2021], PIQA [Bisk et al., 2020], BoolQ [Clark et al., 2019], OpenBookQA [Mihaylov et al., 2018], LAMBADA [Paperno et al., 2016], and Social IQA [Sap et al., 2019]. All zero-shot evaluations are conducted via the lm-eval-harness framework [Gao et al., 2024]. Table 1 reports mean$\pm$std over three calibration seeds (0, 42, 123); all other experiments use a single calibration seed (seed 42) unless the caption explicitly states otherwise. To scale FPoT training to 14B models on a single A100-80GB GPU, we implement a fused Triton [Tillet et al., 2019] kernel that merges the FPoT quantize–dequantize–STE pipeline into a single launch, reducing per-projection intermediate memory by 82%.

5.2 Main Results

Table 1 presents the main comparison of our proposed FPoT method against the non-PoT (uniform INT) baseline and standard PoT across seven models and three bit-width configurations. Our default configuration uses FPoT with $G{=}16$ grid-alignment and 1-adder approximate-grid quantization ($\sqrt{2} \approx 1.5$); activations and KV cache use uniform INT quantization throughout.

FPoT achieves near-lossless PoT quantization in 20 of 21 settings (per the ΔPPL$<+0.5$, Δ0-shot$^9 \geq -1.0$%p criterion stated in the introduction). On Llama-3-8B W4A4KV4, FPoT reduces PPL overhead from $+1.43$ (PoT) to $+0.24$—an 83% reduction—while downstream accuracy degradation shrinks from -5.6%p to -0.4%p. Since $k{=}2$ is specifically optimized for the 4-bit regime (§4.1), we analyze results primarily within the W4 operating points (W4A4KV4 and W4A8KV8, 14 configurations across seven models): ΔPPL ranges from $+0.05$ (Qwen2.5-14B W4A4KV4) to $+0.26$ (Qwen2.5-1.5B W4A8KV8)—*strictly below* $+0.3$ *at the designed operating point*—with PoT overhead reductions of 71%–94% (67%–94% across all twenty-one configurations including W8A8KV8). The W8A8KV8 configurations lie outside the 4-bit-specific design range yet still satisfy the $+0.5$ near-lossless threshold (ΔPPL up to $+0.42$), demonstrating that FPoT$_{k=2}$ generalizes beyond its design point.

5.3 $\sqrt{2}$ Approximation Quality

The FPoT PE requires a fixed $\times\sqrt{2}$ for odd-exponent weights ($\sim 50\%$ of MACs). Table 2 compares canonical signed digit (CSD) shift-add approximations on Llama-3-8B (W4A4KV4) under three strategies: naïve post-hoc substitution ("Exact Grid": quantize to exact grid, dequantize with approximate coefficient), *approximate-grid quantization* ("Approx Grid": calibrate on the actual hardware grid), and *Aware* training (use the approximate $\sqrt{2}$ during both OSTQuant training and GPTQ/eval). CU cost is evaluated under the same naïve component model as §4.2, revealing that additional CSD adders rapidly consume the FPoT hardware advantage: the 3-adder variant already matches INT4 cost ($1.00\times$), leaving only the 1-adder at the knee of the Pareto frontier.

With naïve substitution, the 1-adder ($y + y{\gg}1 = 1.5y$ in Fig. 1(d), 6.1% error) degrades PPL by $+0.17$. However, with approximate-grid quantization, the same 1-adder achieves PPL$= 7.574$—within $+0.01$ of exact $\sqrt{2}$—because weight calibration absorbs the approximation error as part of the quantization grid rather than as post-hoc distortion. The hierarchy **Approx Grid** $>$ **Aware** $>$

Table 1: Main results across models and quantization configurations (mean$_{\pm\text{std}}$ over 3 calibration seeds). Uniform: OSTQuant uniform INT baseline; PoT (base 2) and FPoT (base $\sqrt{2}$, Ours) replace the weight grid on the same pipeline; default configuration in §5.2. PPL: WikiText-2 perplexity (↓). 0-shot^9: mean zero-shot accuracy across nine benchmarks (↑, %).

		W4A4KV4		W4A8KV8		W8A8KV8	
Model	Weight Quant	PPL↓	0-shot^9↑	PPL↓	0-shot^9↑	PPL↓	0-shot^9↑
Llama-3-8B	Uniform	$7.36_{\pm.07}$	$65.2_{\pm.4}$	$6.60_{\pm.01}$	$67.5_{\pm.1}$	$6.16_{\pm.00}$	$68.2_{\pm.1}$
	PoT	$8.79_{\pm.04}$	$59.6_{\pm.6}$	$7.64_{\pm.02}$	$64.0_{\pm.7}$	$7.52_{\pm.12}$	$63.8_{\pm.7}$
	FPoT (Ours)	$\mathbf{7.60}_{\pm.02}$	$\mathbf{64.8}_{\pm.6}$	$\mathbf{6.85}_{\pm.01}$	$\mathbf{67.3}_{\pm.5}$	$\mathbf{6.58}_{\pm.00}$	$\mathbf{67.7}_{\pm.2}$
Llama-2-7B	Uniform	$5.97_{\pm.01}$	$62.1_{\pm.3}$	$5.63_{\pm.04}$	$63.8_{\pm.2}$	$5.48_{\pm.00}$	$65.1_{\pm.0}$
	PoT	$6.93_{\pm.00}$	$58.0_{\pm.1}$	$6.27_{\pm.03}$	$61.4_{\pm.7}$	$6.05_{\pm.04}$	$62.3_{\pm.2}$
	FPoT (Ours)	$\mathbf{6.11}_{\pm.04}$	$\mathbf{61.9}_{\pm.2}$	$\mathbf{5.75}_{\pm.02}$	$\mathbf{63.8}_{\pm.2}$	$\mathbf{5.65}_{\pm.03}$	$\mathbf{64.2}_{\pm.3}$
Llama-2-13B	Uniform	$5.26_{\pm.01}$	$65.6_{\pm.2}$	$5.03_{\pm.01}$	$67.1_{\pm.3}$	$4.89_{\pm.00}$	$67.7_{\pm.0}$
	PoT	$5.90_{\pm.02}$	$62.7_{\pm.5}$	$5.48_{\pm.01}$	$65.2_{\pm.3}$	$5.31_{\pm.03}$	$65.2_{\pm.7}$
	FPoT (Ours)	$\mathbf{5.37}_{\pm.02}$	$\mathbf{65.2}_{\pm.1}$	$\mathbf{5.11}_{\pm.02}$	$\mathbf{66.6}_{\pm.1}$	$\mathbf{5.03}_{\pm.01}$	$\mathbf{67.2}_{\pm.4}$
Qwen2.5-1.5B	Uniform	$11.10_{\pm.14}$	$38.7_{\pm.4}$	$9.75_{\pm.03}$	$42.7_{\pm.6}$	$9.32_{\pm.00}$	$44.0_{\pm.2}$
	PoT	$13.23_{\pm.11}$	$37.2_{\pm.8}$	$11.35_{\pm.09}$	$39.3_{\pm.4}$	$11.33_{\pm.05}$	$38.8_{\pm1.4}$
	FPoT (Ours)	$\mathbf{11.23}_{\pm.03}$	$\mathbf{39.9}_{\pm.4}$	$\mathbf{10.01}_{\pm.03}$	$\mathbf{42.4}_{\pm1.0}$	$\mathbf{9.73}_{\pm.01}$	$\mathbf{43.0}_{\pm.7}$
Qwen2.5-7B	Uniform	$7.56_{\pm.01}$	$66.4_{\pm.5}$	$7.04_{\pm.01}$	$68.1_{\pm.1}$	$6.87_{\pm.00}$	$68.6_{\pm.1}$
	PoT	$8.22_{\pm.01}$	$62.9_{\pm.6}$	$7.56_{\pm.02}$	$65.5_{\pm.2}$	$7.57_{\pm.02}$	$66.2_{\pm1.0}$
	FPoT (Ours)	$\mathbf{7.62}_{\pm.02}$	$\mathbf{66.3}_{\pm.3}$	$\mathbf{7.14}_{\pm.01}$	$\mathbf{68.6}_{\pm.6}$	$\mathbf{7.03}_{\pm.02}$	$\mathbf{68.5}_{\pm.6}$
Qwen2.5-14B	Uniform	$6.33_{\pm.01}$	$68.7_{\pm.4}$	$5.65_{\pm.01}$	$71.1_{\pm.4}$	$5.31_{\pm.00}$	$71.1_{\pm.2}$
	PoT	$7.21_{\pm.05}$	$64.8_{\pm1.1}$	$6.48_{\pm.05}$	$68.8_{\pm.3}$	$6.40_{\pm.01}$	$68.7_{\pm.4}$
	FPoT (Ours)	$\mathbf{6.38}_{\pm.04}$	$\mathbf{68.9}_{\pm.3}$	$\mathbf{5.85}_{\pm.08}$	$\mathbf{71.3}_{\pm.8}$	$\mathbf{5.60}_{\pm.00}$	$\mathbf{72.0}_{\pm.1}$
Mistral-7B	Uniform	$5.62_{\pm.01}$	$66.5_{\pm.3}$	$5.41_{\pm.02}$	$67.8_{\pm.1}$	$5.32_{\pm.00}$	$68.3_{\pm.1}$
	PoT	$6.00_{\pm.03}$	$64.6_{\pm.4}$	$5.69_{\pm.02}$	$65.5_{\pm.7}$	$5.71_{\pm.02}$	$66.4_{\pm.3}$
	FPoT (Ours)	$\mathbf{5.71}_{\pm.05}$	$\mathbf{66.1}_{\pm.5}$	$\mathbf{5.49}_{\pm.04}$	$\mathbf{67.3}_{\pm.4}$	$\mathbf{5.45}_{\pm.07}$	$\mathbf{67.7}_{\pm.4}$

Table 2: $\sqrt{2}$ approximation strategies on Llama-3-8B W4A4KV4 (FPoT; single calibration seed). Exact Grid / Approx Grid (Ours) / Aware: see §5.3 for definitions. CU/×: same naïve component model (§4.2); ‡exact $\sqrt{2}$ has no finite CSD realization. 0-shot^9 in %.

				Exact Grid		Approx Grid		Aware	
Impl.	Value (Err.)	Add.	CU/×	PPL↓	0-shot^9↑	PPL↓	0-shot^9↑	PPL↓	0-shot^9↑
Exact	1.414 (0%)	—	N/A‡	7.560	64.7	—	—	—	—
3-adder	1.438 (1.6%)	3	1.00×	7.580	64.4	7.569	65.0	7.611	64.3
2-adder	1.375 (2.8%)	2	0.75×	7.575	64.5	7.584	64.4	7.584	64.8
1-adder	**1.500 (6.1%)**	**1**	**0.50×**	7.729	63.4	**7.574**	**64.2**	7.625	63.8
k=1	—	0	0.38×	8.806	59.0	—	—	—	—

‡ Exact $\sqrt{2}$ requires a full multiplier-grade unit (no finite CSD realization); not a shift-add alternative.

Exact Grid holds for the 1-adder case (PPL $7.574 < 7.625 < 7.729$): aware training (7.625) lies between the two extremes since the STE gradient through approximate dequantization is less accurate than through exact dequantization, while smaller approximation errors (2-adder, 3-adder) make all three strategies converge. This yields a simple PE whose compute unit is $\sim 0.50\times$ that of an INT4 multiplier with no measurable accuracy loss—a Pareto-optimal point among CSD variants (Table 2): the 2-adder and 3-adder alternatives pay $1.5\times$ and $2\times$ more CU, respectively, for at most a 0.005 PPL improvement, while the 0-adder PoT ($0.38\times$) incurs an unusable $+1.23$ PPL penalty.

5.4 Additional Experiments

Preprocessing. Both rotation and per-channel scaling are necessary: without any transform FPoT yields catastrophic PPL $= 181.3$; only the full combination (learned rotation + scale) reaches 7.56, within $+0.12$ of the non-PoT baseline (Table 13, Appendix G.6).

Preprocessing $\times$ quantizer factorial. Training the preprocessing for one grid and quantizing to another separates the two factors. The weight quantizer moves PPL about ten times more than the preprocessing target (1.51 versus 0.15 across means), and the order INT, FPoT, PoT holds under every preprocessing (Appendix K).

Full FPoT (weight + activation). Applying the logarithmic grid to both weights and activations degrades accuracy severely (PPL 16.9–114.1 at k=1, 8.21–52.49 at k=2, versus 7.56 for weight-only; Appendix G.3). Uniform INT activations remain the practical choice.

Weight grid-alignment. Grid-alignment (Eq. (4)) reduces PPL by up to -0.051 at k=1 at zero inference cost; at k=2 the finer grid self-corrects ($|\Delta\text{PPL}| \leq 0.034$; Appendix G.2).

Hyperparameter sensitivity. The zero threshold τ and log-vs-linear rounding mode have negligible impact ($\Delta\text{PPL} < 0.03$); see Appendices G.4 and G.5.

Calibration robustness. FPoT's data-independent grid boundaries yield robust calibration: $|\Delta\text{PPL}| \leq 0.065$ across $n \in \{128, 256, 512\}$ and only $+0.06$–$+0.10$ shift from WikiText-2 to C4 (Appendix G.1).

Cross-framework portability. A different PTQ framework yields comparable FPoT overhead ($+0.05$–$+0.34$ PPL; Appendix J, Table 22).

Generation tasks. On two instruction-tuned models, standard PoT trails uniform INT by about 15 to 25 points on math and code, and FPoT recovers most of that gap. On Llama-3.1-8B-Instruct, for example, HumanEval rises from 27.6 with PoT to 47.8 with FPoT, against 52.4 for INT (Appendix L).

LogART comparison. In the weight-only setting (W4A16), LogART achieves lower PPL but degrades catastrophically (PPL > 3900) under post-hoc INT4 activation without PTQ preprocessing, while FPoT remains stable (Appendix I, Table 21).

6 Discussion

Design rationale and comparison with prior PoT methods. FPoT's three design choices—a regular geometric grid amenable to GPTQ (vs. APoT's irregular additive decompositions [Li et al., 2020]), uniform INT activations eliminating runtime $\log_2$ conversion [Elhoushi et al., 2021, Li et al., 2023], and 4-bit grid-resolution targeting (vs. PoT-PTQ's 2–3-bit scale-calibration focus [Wang et al., 2025])—together let FPoT reach near-lossless W4A4KV4 with a multiplier-free datapath. LogART [Xu et al., 2026] reports strong W4A16 accuracy but lacks LLM W4A4KV4 evaluation due to absent PTQ preprocessing (Appendix I).

Broader impact. Multiplier-free LLM inference can reduce the energy consumption and carbon footprint of large-scale deployment. However, cheaper inference may also lower barriers to harmful applications; this work does not introduce new capabilities and applies to existing public models with slightly degraded accuracy.

Limitations. Efficiency projections use analytical cost models anchored to published RTL synthesis [Przewlocka-Rus et al., 2022]; open-source logic synthesis (Yosys [Wolf and Glaser, 2013]) confirms the structural ordering and bit-width scaling (Appendix E.2), and end-to-end RTL/ASIC validation is left for future work. These figures are per weight MAC, so the end-to-end gain is largest in compute-bound prefill and smaller in memory-bound decode (Appendix D). FPoT depends on PTQ preprocessing. Without it, W4A4KV4 FPoT reaches PPL 181.3 (Appendix G.6). Uniform INT at W4A4KV4 also relies on such preprocessing to handle activation outliers [Ashkboos et al., 2024, Liu et al., 2025], and the factorial in Appendix K shows that FPoT keeps its advantage over PoT under

each preprocessing target. Preprocessing training requires float32 weight copies, limiting single-GPU (80 GB) experiments to $\leq$14B parameters. The k=2 optimum is 4-bit specific (8-bit shifts to $\approx$19).

7 Conclusion

We introduced FPoT—a half-octave fractional PoT weight quantization scheme that, combined with uniform INT activation quantization and state-of-the-art LLM PTQ preprocessing, achieves near-lossless accuracy while enabling multiplier-free shift-add inference. Three contributions enable this result. First, a Gaussian-proxy analysis identifies base-$\sqrt{2}$ as the optimal integer-resolution grid at 4-bit, halving the worst-case relative error of standard PoT ($41.4\%\rightarrow18.9\%$). Second, approximate-grid quantization absorbs the 1-adder $\sqrt{2}$ hardware error via weight calibration, yielding a PE whose compute unit is $\sim0.50\times$ the INT4 multiplier gate count (vs. $\sim0.38\times$ for standard PoT). Third, across twenty-one configurations spanning seven models and three quantization settings (W4A4KV4 / W4A8KV8 / W8A8KV8), FPoT recovers zero-shot accuracy to within 1.0%p of the uniform INT baseline in $20/21$ configurations (mean gap -0.17%p; all 14 W4 configurations within 0.5%p), and integrates with state-of-the-art LLM PTQ frameworks at comparable overhead (Appendix J).

These results suggest that FPoT is a promising pathway toward shift-based LLM inference hardware with near-lossless accuracy, with future work including scaling to larger models, exploring higher bit-widths, and end-to-end RTL/ASIC validation.

Acknowledgments and Disclosure of Funding

This work was supported by the Korean Government through grants from IITP (RS-2021-II211343 and RS-2025-25442338) and NRF (RS-2026-25500579). The authors declare no competing interests.

References

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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439, 2020.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

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. *arXiv preprint arXiv:1803.05457*, 2018.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332, 2022.

Mostafa Elhoushi, Zihao Chen, Farhan Shafiq, Ye Henry Tian, and Joey Yiwei Li. DeepShift: Towards multiplication-less neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2359–2368, 2021.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff,

Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, July 2024. URL https://zenodo.org/records/12608602.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, Zhihang Yuan, Sifan Zhou, et al. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations*, 2025.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Xinlin Li, Bang Liu, Rui Heng Yang, Vanessa Courville, Chao Xing, and Vahid Partovi Nia. DenseShift: Towards accurate and efficient low-bit power-of-two quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17010–17020, 2023.

Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations*, 2020.

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.

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.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, 2016.

Dominika Przewlocka-Rus, Syed Shakib Sarwar, H. Ekin Sumbul, Yuecheng Li, and Barbara De Salvo. Power-of-two quantization for low bitwidth and hardware compliant neural networks. In *Proceedings of the tinyML Research Symposium*, San Jose, CA, 2022.

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, 2019.

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

Philippe Tillet, H. T. Kung, and David Cox. Triton: An intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages (MAPL)*, pages 10–19, 2019. doi: 10.1145/3315508.3329973.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Xinyu Wang, Vahid Partovi Nia, Peng Lu, Jerry Huang, Xiao-Wen Chang, Boxing Chen, and Yufei Cui. PoT-PTQ: Two-step power-of-two post-training for LLMs. In *European Conference on Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2025. doi: 10.3233/FAIA251188.

Clifford Wolf and Johann Glaser. Yosys – a free Verilog synthesis suite. In *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip)*, pages 1–6, 2013.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023.

Jiawei Xu, Yuxiang Huan, Li-Rong Zheng, and Zhuo Zou. A low-power arithmetic element for multi-base logarithmic computation on deep neural networks. In *IEEE International System-on-Chip Conference (SOCC)*, pages 43–48, 2018. doi: 10.1109/SOCC.2018.8618560.

Jiawei Xu, Jiangshan Fan, Baolin Nan, Chen Ding, Li-Rong Zheng, Zhuo Zou, and Yuxiang Huan. ASLog: An area-efficient CNN accelerator for per-channel logarithmic post-training quantization. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(12):5380–5393, 2023. doi: 10.1109/TCSI.2023.3315299.

Jiawei Xu, Yi Zheng, Chenghe Sun, Taiyu Zhou, Zuqi Zhang, Jie Li, Lirong Zheng, and Zhuo Zou. LogART: Pushing the limit of efficient logarithmic post-training quantization. In *The Fourteenth International Conference on Learning Representations*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

A Preliminaries

A.1 Uniform Quantization

Standard uniform (linear) quantization maps a floating-point tensor to a finite set of evenly-spaced levels. For symmetric b-bit quantization with per-token dynamic scale s:

$$Q_{\text{lin}}(x) = \text{clamp}\Big(\Big\lfloor \frac{x}{s} \Big\rceil, -2^{b-1}, 2^{b-1}-1\Big) \cdot s, \tag{6}$$

where $s = \|x\|_\infty/(2^{b-1}-1)$. In this work, activations and the KV cache use uniform INT quantization in all configurations (in its asymmetric form, with a zero point), while weights use logarithmic (FPoT) quantization (§3).

A.2 Power-of-Two Quantization

PoT quantization restricts each weight to zero or $\pm 2^n$ times a scale. The product of an integer activation x with 2^n is a barrel shift of x by n bits, so a MAC becomes a shift followed by an integer addition into the accumulator and needs no multiplier array. The idea was developed for CNNs. DeepShift [Elhoushi et al., 2021] trains networks whose weights are single signed powers of two. APoT [Li et al., 2020] represents each weight as a sum of n PoT terms and pays one shift-add per term. DenseShift [Li et al., 2023] removes the zero level so that every code carries a nonzero value. Base-$\sqrt{2}$ grids appear in logarithmic accelerators, which approximate $\sqrt{2}$ by $2^0 + 2^{-1}$ [Xu et al., 2018, 2023], and in LogART [Xu et al., 2026]. With one level per octave, a 4-bit PoT grid rounds with up to 41.4% relative error. CNNs trained or fine-tuned for the grid tolerate this, but LLM weights quantized after training do not. Standard PoT adds $+1.43$ PPL on Llama-3-8B W4A4KV4 even with preprocessing (Table 1). FPoT halves the octave, so one extra adder buys a twice finer grid.

A.3 State-of-the-Art LLM PTQ Preprocessing

Recent LLM PTQ preprocessing methods (QuaRot [Ashkboos et al., 2024], SpinQuant [Liu et al., 2025], OSTQuant [Hu et al., 2025], DuQuant [Lin et al., 2024a]; see Section 2 for taxonomy) apply learned transformations—orthogonal rotations, per-channel scaling, or their composition—to redistribute channel-concentrated outliers before quantization. For input activation $X \in \mathbb{R}^{n\times d}$ and weight $W \in \mathbb{R}^{d\times m}$, these methods rewrite the linear operation as

$$Y = XW = (XT)(T^{-1}W) = \hat{X}\hat{W}, \tag{7}$$

where T is an invertible transform (orthogonal rotation, scaling matrix, or their composition). This reduces kurtosis and produces smoother distributions amenable to the logarithmic PoT grid. We primarily build on OSTQuant, which jointly trains a Hadamard rotation with per-channel smooth scales via Riemannian gradient descent on the Stiefel manifold (400 steps with STE). The transform adds only $\approx 0.02\%$ overhead in MAC operations ($\sim$49 K additions per layer at d=4,096 vs. 218 M weight MACs). Our FPoT method is compatible with any such preprocessing; we primarily evaluate with OSTQuant and confirm DuQuant portability in Appendix J.

B Grid-Matching Theory

B.1 Normalized Bin Entropy: Properties and Proofs

We establish five properties of Normalized Bin Entropy (NBE, Eq. (5)) that justify its use as a grid-level design metric.

Property 1 (Bounds). For any probability vector $\mathbf{p} \in \Delta^{N-1}$, $0 \le \text{NBE}(\mathbf{p}) \le 1$, with $\text{NBE} = 1$ iff $p_i = 1/N$ for all i (uniform occupancy), and $\text{NBE} = 0$ iff $p_i = 1$ for some single level i.

Proof. By the non-negativity of Shannon entropy, $H(\mathbf{p}) \ge 0$ with equality iff $\mathbf{p}$ is a point mass. The upper bound $H(\mathbf{p}) \le \log_2 N$ is the maximum-entropy property of the uniform distribution over N outcomes, proven by Jensen's inequality applied to $-x \log x$. Dividing by $\log_2 N$ yields the stated bounds. □

Property 2 (Information-theoretic interpretation). Let $L^\star(\mathbf{p})$ denote the expected code length of an optimal prefix-free (Shannon–Fano or Huffman) code for the occupancy distribution $\mathbf{p}$. Then

$$\log_2 N \cdot \mathrm{NBE}(\mathbf{p}) \;=\; H(\mathbf{p}) \;\le\; L^\star(\mathbf{p}) \;<\; H(\mathbf{p}) + 1.$$

That is, $\log_2 N \cdot \mathrm{NBE}$ is the tightest achievable expected code length per weight, and NBE measures the fraction of the fixed-length budget $\log_2 N$ that carries information under the data distribution.

Proof. Shannon's source coding theorem states that for any distribution $\mathbf{p}$, $H(\mathbf{p}) \le L^\star(\mathbf{p}) < H(\mathbf{p})+1$ bits per symbol. Dividing by $\log_2 N$ and rearranging yields the claim. □

Property 3 (Affine invariance of level values). NBE depends only on the occupancy probabilities $\mathbf{p}$, not on the level values $\{\ell_i\}$. Consequently, affine relabeling $\ell'_i = a\ell_i + b$ ($a \neq 0$) leaves NBE unchanged, provided the induced decision boundaries assign the same weight population to each level.

Proof. Eq. (5) depends only on $\mathbf{p}$. Under affine relabeling with unchanged decision topology, each p_i is preserved. □

This property ensures that NBE captures the *shape* of the level-mass mapping independently of level magnitudes, distinguishing it from value-dependent metrics such as MSE.

Property 4 (Grid-structure sensitivity under fixed dynamic range). Let Q_1, Q_2 be two quantizers with identical N and identical dynamic range $[-R, R]$ but different level placements $\{\ell_i^{(1)}\}, \{\ell_i^{(2)}\}$ (e.g., uniform vs. logarithmic). For a non-uniform weight distribution p_W concentrated near 0, generally $\mathrm{NBE}(Q_1; p_W) \neq \mathrm{NBE}(Q_2; p_W)$.

Proof sketch. Different level placements yield different decision regions, which partition p_W into different mass vectors $\mathbf{p}^{(1)} \neq \mathbf{p}^{(2)}$. While Shannon entropy can coincidentally coincide on distinct probability vectors (e.g., under permutation), structurally different partitions of a log-concave p_W—such as those induced by uniform vs. logarithmic grids—yield mass vectors that differ not merely by permutation but in their degree of concentration, producing distinct entropies. For $\mathcal{N}(0,1)$ weights with N=15, uniform INT4 yields NBE $= 0.9026$ while FPoT$_{k=2}$ yields NBE $= 0.9364$ (Table 3), quantitatively confirming this claim. □

This is the property that volumetric utilization metrics (e.g., QSUR [Hu et al., 2025]) lack: they depend only on the bounding hypercube volume, not on the internal level structure.

Property 5 (Correlation with MSE for log-concave distributions). For log-concave symmetric weight distributions (satisfied by the Gaussian proxy and, after PTQ preprocessing, empirically by LLM weights), the quantizer base $\beta = 2^{1/k}$ that maximizes NBE lies within a narrow neighborhood of the base that minimizes MSE. Specifically, for $\mathcal{N}(0,1)$ weights with 4-bit budget ($N = 15$), both optima satisfy $\beta^\star \in [2^{0.50}, 2^{0.60}]$ (equivalently, $k^\star \in [1.67, 2.00]$).

Empirical justification. The MSE of a log-base-β quantizer on p_W decomposes into a tail-clipping term (decreasing in β, as a larger β^{N-1} reduces clipped tail mass) and an in-range rounding term (increasing in β, as coarser bins accept larger within-bin errors); the interior optimum balances these effects. NBE exhibits the analogous trade-off: too-small β concentrates mass in tail-proximal bins, too-large leaves high-density bins over-populated. Numerical sweep on $\mathcal{N}(0,1)$ with N=15 confirms both optima in the stated interval (Table 3, Fig. 2). A formal rate-distortion analysis is left to future work.

Remark. Properties 1–4 are exact; Property 5 is empirical for our operating regime and serves to justify why NBE-based grid design yields near-MSE-optimal quantizers in practice.

B.2 Log-Magnitude Distribution Derivation

Let $u = \log_2 |w|$ for $w \sim \mathcal{N}(0, \sigma^2)$. Its probability density is

$$f_U(u) = \frac{2 \ln 2}{\sqrt{2\pi}\,\sigma}\, 2^u \, \exp\Big(-\frac{2^{2u}}{2\sigma^2}\Big). \tag{8}$$

Substituting $v = u - \log_2 \sigma$ and expanding near v=0 gives $f_U \propto \exp(-v^2)$, so the log-magnitude is approximately Gaussian in log-space with standard deviation $1/(\sqrt{2}\ln 2) \approx 1.02$ octaves ($2^{1.02}$).

Table 3: MSE and NBE of 4-bit PoT quantizers applied to $\mathcal{N}(0,1)$ weights (N=15 levels: 14 non-zero symmetric + 1 zero, log-domain rounding). MSE ratio is relative to uniform INT4.

Method	MSE	NBE	MSE ratio
Uniform INT4	0.0128	0.9026	1.00×
PoT	0.0411	0.8773	3.21×
FPoT$_{k=2}$	**0.0162**	**0.9364**	**1.27×**
FPoT$_{k=3}$	0.0368	0.8762	2.87×
FPoT$_{k=4}$	0.0688	0.8115	5.37×
Optimal ($k^\star$≈1.8)	0.0158	0.9417	1.23×

Consequently, ∼95% of probability mass concentrates within ∼4 octaves around the mode. This ∼4-octave concentration is not unique to exact Gaussians: any symmetric, outlier-suppressed distribution with comparable kurtosis exhibits a similar log-domain spread, which is the condition that PTQ preprocessing enforces.

B.3 MSE and NBE Across Integer k

Table 3 reports the 4-bit MSE and NBE under the Gaussian proxy across k. The optimal base $\beta^\star \approx 2^{0.55}$ yields an effective $k^\star \approx 1.8$ (MSE $= 1.23\times$ INT4, NBE $= 0.9417$); integer k=2 is the closest practical choice.

Bit-width dependence. At 8-bit (127 magnitudes per side), the abundance of levels permits much finer resolution without dynamic-range starvation, shifting the optimal to $k^\star \approx 19$. FPoT$_{k=2}$ thus does not generalize as the preferred grid at higher bit-widths.

C Empirical Bin Occupancy

While the Gaussian-proxy bin occupancy (Fig. 2, §4.1) establishes the grid-matching mechanism analytically, we validate this on real LLM weights here. Fig. 4 compares the quantization grids of uniform INT4, PoT, and FPoT on the number line. Fig. 5 shows the aggregated empirical bin occupancy for Llama-3-8B (W4A4KV4).

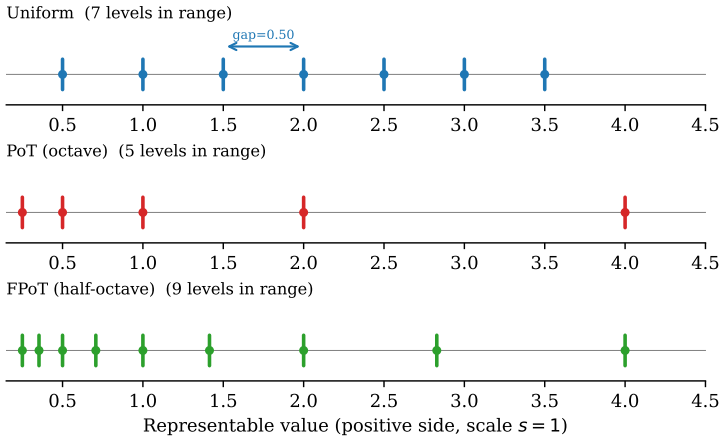


Figure 4: Quantization grid comparison: Uniform, PoT (octave), and FPoT (half-octave). The FPoT grid places 2× more levels within the high-density region of the Gaussian weight distribution.

We present the bin occupancy distributions of RTN-quantized weights (W4A4KV4) for each model evaluated in this work. These empirical distributions complement the theoretical Gaussian analysis (Fig. 2) and the Llama-3-8B aggregated results (Fig. 5).

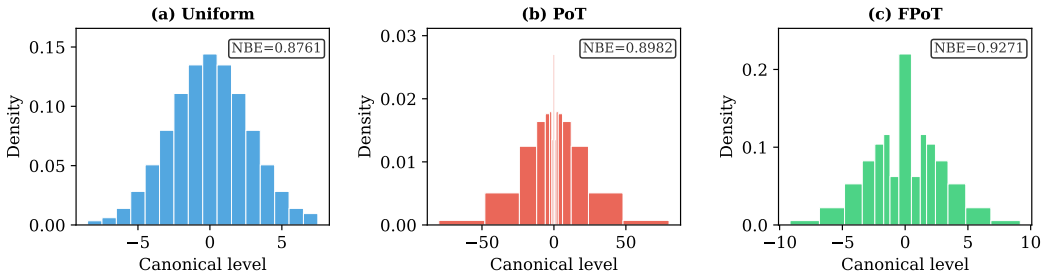


Figure 5: Empirical bin occupancy for Llama-3-8B (W4A4KV4, aggregated across all projections). The pattern closely matches the Gaussian proxy prediction in Fig. 2.

Across all models, the empirical bin occupancy closely matches the theoretical Gaussian prediction: (a) uniform INT4 exhibits a bell-shaped occupancy with $\sim$14% zero bin; (b) PoT concentrates weights in the mid-exponent bins (± 16, ± 32) with only $\sim$3% zero; (c) FPoT achieves more uniform non-zero bin occupancy but maps $\sim$20–22% of weights to zero due to the k-adaptive zero threshold $\tau_2 = 2^{-1/4} \approx 0.841$ (Section 3). The consistency across model families (Llama-2, Llama-3, Qwen2.5) and scales (1.5B–14B) confirms that recent LLM PTQ preprocessing produces weight distributions with similar log-domain concentration, validating the use of the Gaussian proxy analysis for grid design.

C.1 Per-Projection Breakdown

Fig. 6 shows the per-projection bin occupancy for Llama-3-8B, complementing the aggregated view in Fig. 5.

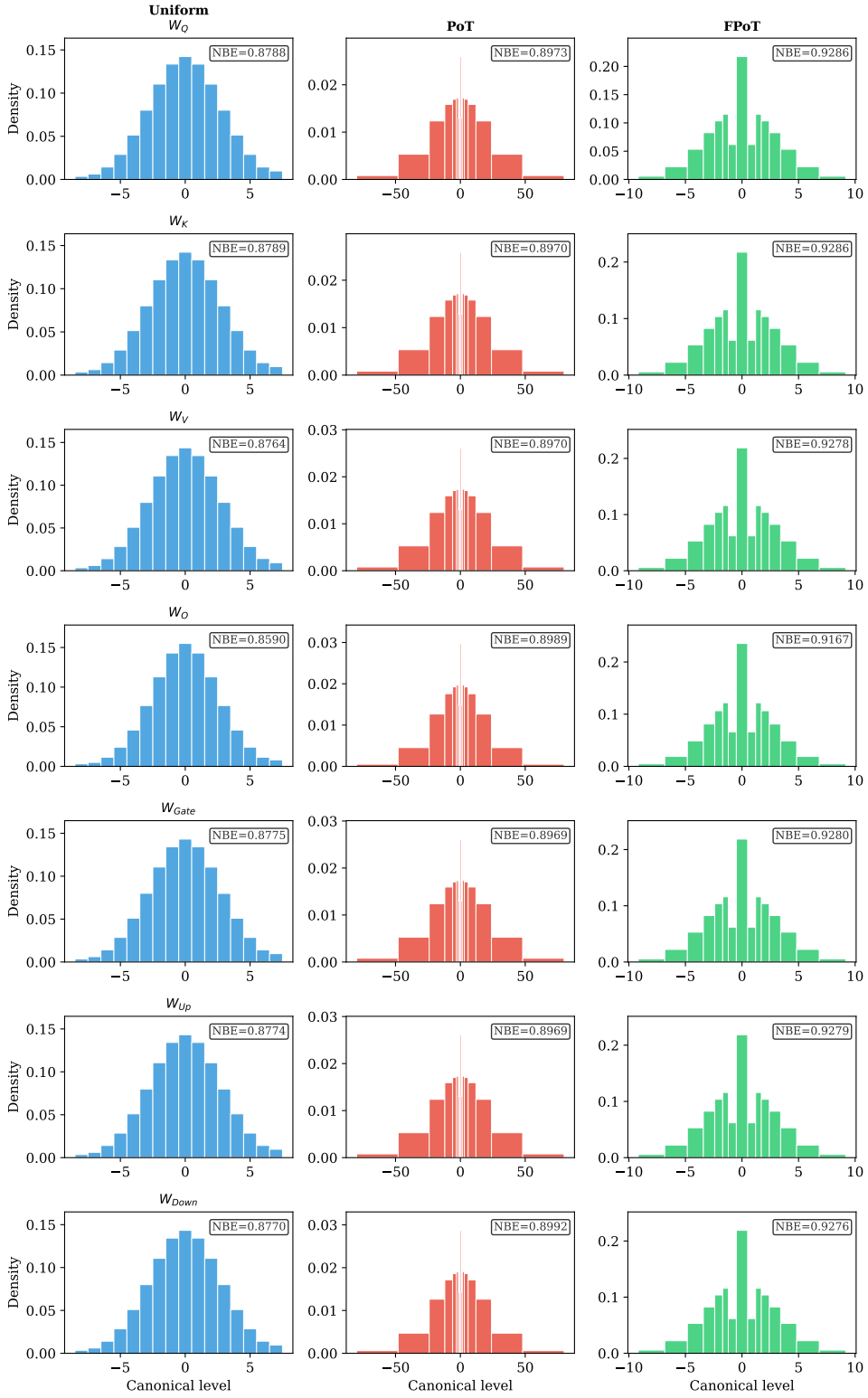


Figure 6: Per-projection bin occupancy of RTN-quantized Llama-3-8B weights (W4A4KV4). Rows: W_Q, W_K, W_V, W_O, W_{Gate}, W_{Up}, W_{Down}. Columns: Uniform, PoT, FPoT. The occupancy pattern is consistent across all seven projections, with FPoT achieving NBE > 0.91 uniformly (0.9167–0.9286), substantially above PoT and uniform INT4.

C.2 Effect of PTQ Preprocessing on Bin Occupancy

Fig. 7–12 compare the bin occupancy *before* and *after* PTQ preprocessing. A consistent pattern emerges: preprocessing increases NBE for uniform INT4 ($+0.01$ to $+0.07$) and FPoT ($+0.01$ to $+0.06$), while slightly *decreasing* NBE for PoT (-0.01 to -0.03). This confirms that distribution redistribution specifically benefits grids whose dynamic range is matched to the redistributed weight spread ($\sim$4 octaves), while the excessively wide PoT grid (64:1) cannot exploit the narrower post-preprocessing distribution. The effect is most pronounced for Qwen2.5 models, whose raw weights exhibit stronger outlier concentration.

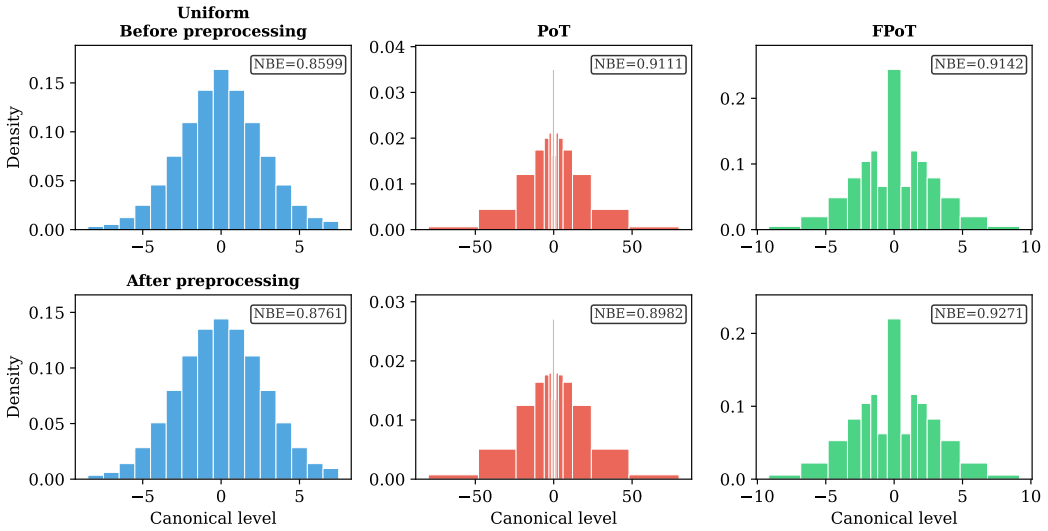


Figure 7: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Llama-3-8B (W4A4KV4). uniform INT4 NBE rises from 0.8599 to 0.8761 and FPoT from 0.9142 to 0.9271; PoT drops slightly (0.9111 $\rightarrow$ 0.8982).

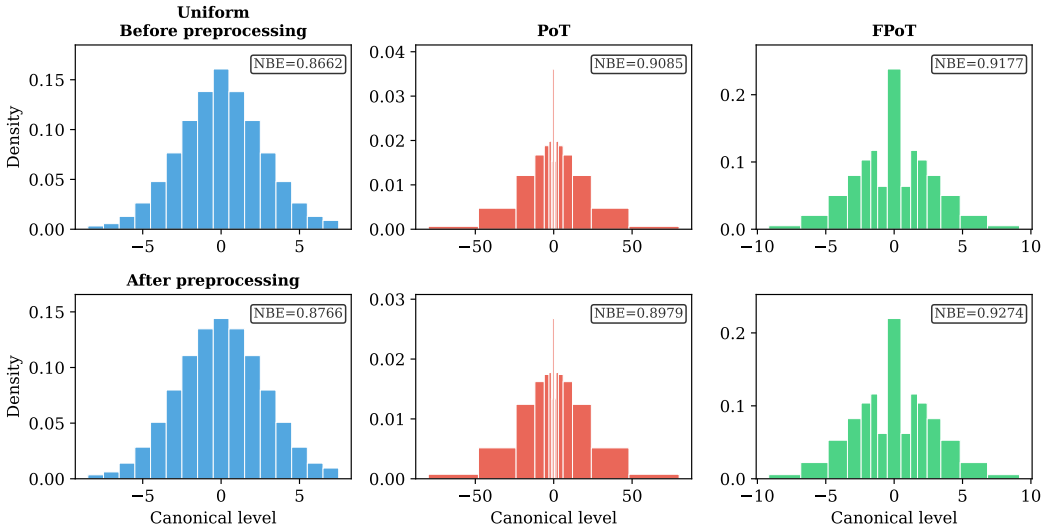


Figure 8: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Llama-2-7B (W4A4KV4). NBE shifts: uniform INT4 0.8662 $\rightarrow$ 0.8766; FPoT 0.9177 $\rightarrow$ 0.9274.

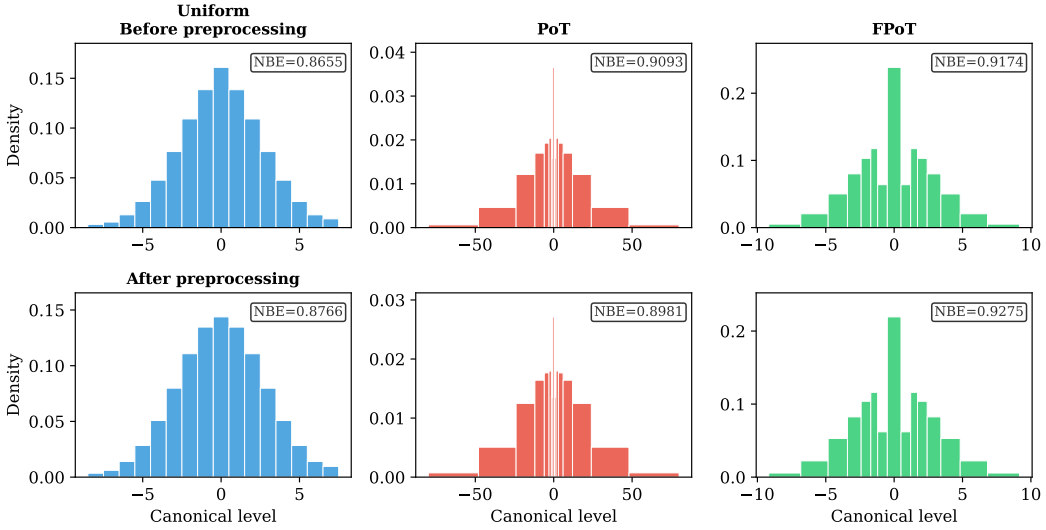


Figure 9: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Llama-2-13B (W4A4KV4). NBE shifts: uniform INT4 0.8655 → 0.8766; FPoT 0.9174 → 0.9275.

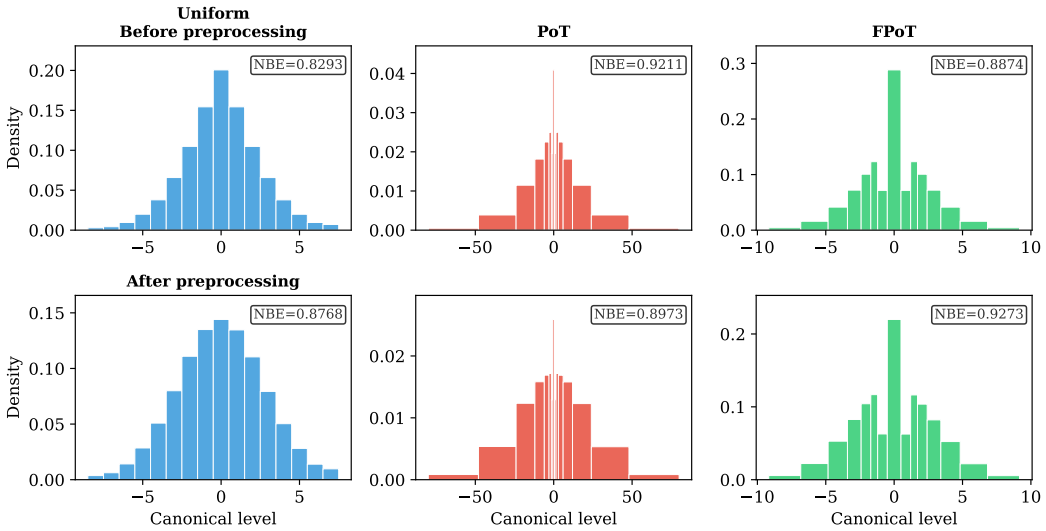


Figure 10: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Qwen2.5-1.5B (W4A4KV4). NBE shifts are large (uniform INT4 0.8293 → 0.8768; FPoT 0.8874 → 0.9273), reflecting strong raw-weight outlier concentration.

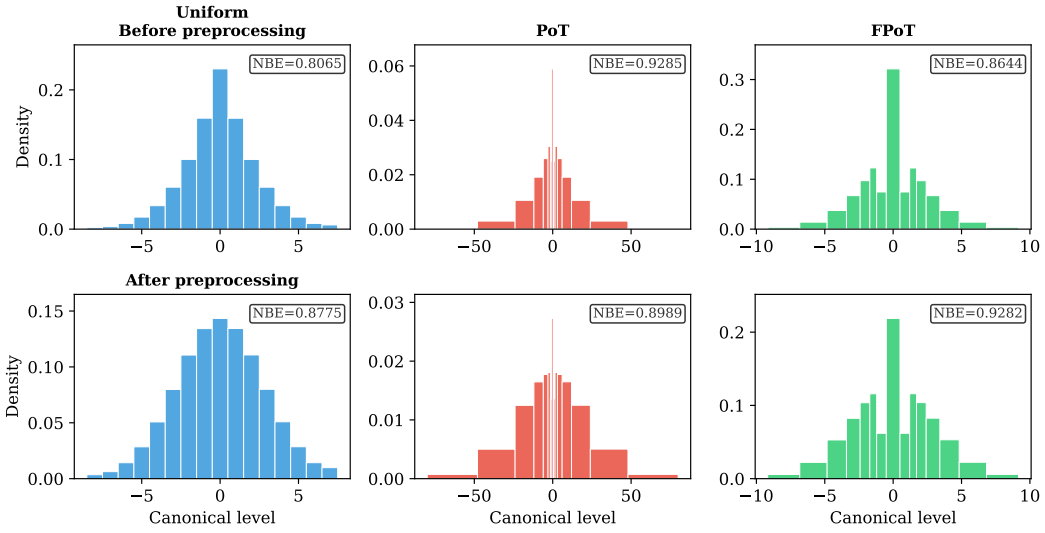


Figure 11: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Qwen2.5-7B (W4A4KV4). NBE shifts: uniform INT4 0.8065 → 0.8775; FPoT 0.8644 → 0.9282.

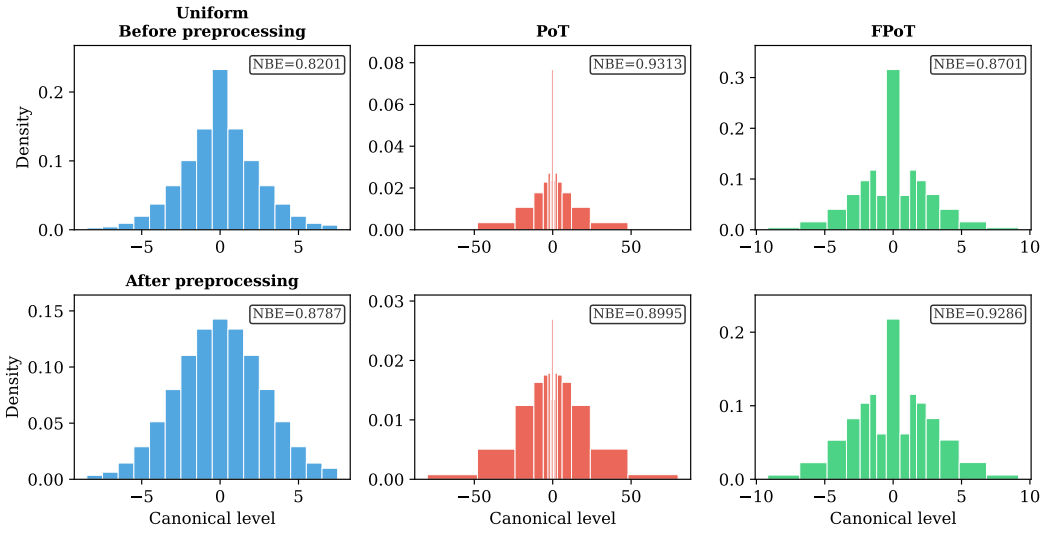


Figure 12: Bin occupancy before (top) and after (bottom) PTQ preprocessing for Qwen2.5-14B (W4A4KV4). NBE shifts: uniform INT4 0.8201 → 0.8787; FPoT 0.8701 → 0.9286.

D Per-Projection Sensitivity Analysis

D.1 Per-Projection MAC Breakdown

Table 4 presents the per-projection MAC breakdown for Llama-3-8B (32 layers, single token); the FFN projections (Gate, Up, Down) dominate at 80.8% of weight MACs.

Table 4: MAC Breakdown per Token for Llama-3-8B (32 Layers)

Projection	MACs (M)	% of Total
W_Q	537	7.7%
W_K	134	1.9%
W_V	134	1.9%
W_O	537	7.7%
Gate	1,879	26.9%
Up	1,879	26.9%
Down	1,879	26.9%
Attn total	*1,342*	*19.2%*
FFN total	*5,637*	*80.8%*
Total	**6,979**	**100%**

Prefill versus decode. The shares above count weight MACs only, and the unquantized LM head is not counted. Attention adds $2d_{\text{model}}c$ MACs per token and layer for $\text{QK}^\top$ and PV, where c is the number of attended positions ($c{=}L/2$ on average in causal prefill of length L, and the KV length in decode). With this count, linear projections carry 96.3% of all MACs for Llama-3-8B in prefill at $L{=}$2K (93.7–96.9% across the seven models), 86.7% at $L{=}$8K (78.8–88.8%), and 76.5% in decode with an 8K KV cache (65.0–79.9%). Applying the 0.42–$0.46\times$ per-weight-MAC energy of §4.2 to this share and keeping attention MACs at INT cost gives a total-MAC energy of 0.44–$0.48\times$ in 2K prefill and 0.56–$0.59\times$ in decode with an 8K KV cache. In memory-bound decode, arithmetic makes up a smaller share of system energy than in prefill, and weight traffic matches uniform INT4 because both store 4-bit weights.

D.2 Per-Projection FPoT Sensitivity

Table 5 presents the per-projection FPoT sensitivity on Llama-3-8B (W4A4KV4). Each row applies weight-only FPoT to the indicated projection(s) while the rest use uniform INT. The non-PoT baseline achieves PPL $= 7.444$ and 0-shot[9] $= 64.98\%$.

Table 5: Per-projection FPoT weight-only ablation on Llama-3-8B (W4A4KV4, 400 steps, single seed). Each row applies FPoT to the indicated projection(s); all others use uniform INT. Activations use uniform INT4 throughout.

FPoT Applied To	PPL↓	0-shot[9]↑ (%)
None (non-PoT baseline)	7.444	64.98
W_Q only	7.379	65.59
W_K only	7.393	64.77
W_V only	7.487	65.45
W_O only	7.337	65.21
Gate only	7.381	65.45
Up only	7.386	65.77
Down only	7.496	64.59
Attention (W_Q+W_K+W_V+W_O)	7.419	65.13
FFN (Gate+Up+Down)	7.530	64.74
All 7 projections	**7.563**	**64.28**

FPoT achieves near-lossless quantization on every individual projection. All single-projection entries achieve PPL within $+0.06$ of the non-PoT baseline (largest $+0.052$, Down), with 0-shot[9]

within 0.4%p. Even when applied to all seven projections simultaneously, the total overhead is only $+0.12$ PPL and -0.7%p 0-shot[9].

Degradation is not additive. Five projections individually land slightly below the baseline, yet all seven together (PPL $= 7.563$) add $+0.12$, so the single-projection entries rank sensitivity rather than form an error budget.

D.3 Design Implications

Per-projection sensitivity guides future aggressive compression designs. The Value projection requires uniform INT activations: full FPoT on W_V incurs ΔPPL$= +0.30$ versus $+0.04$ for weight-only FPoT (Table 5), a $7\times$ gap driven by post-softmax activation noise. For future full-PoT activation designs, the Output projection is the safest candidate, followed by FFN projections; Value should be last.

E Hardware Efficiency Analysis Details

This appendix documents the methodology underlying the hardware cost estimates in Section 4.2 and presents per-model accuracy–cost trade-offs.

E.1 Naïve Component-Based Cost Model

We adopt a simple component-counting model with two unit costs: $c_{\text{fa}} = 1.0$ (full-adder equivalent) for arithmetic logic, and $c_{\text{mux}} = 0.5 \times c_{\text{fa}}$ for 2-to-1 multiplexer stages (reflecting the lower transistor count of a pass-gate MUX vs. a full adder). All costs are per-bit of the datapath width A (activation bits).

INT multiplier. A $W\times A$ signed multiplier generates $W\times A$ partial products and reduces them through an adder tree. Complexity grows *quadratically*:

$$C_{\text{INT}} = W \times A.$$

PoT barrel shifter. A barrel shifter with $(W-1)$-bit shift control implements $(W-1)$ cascaded MUX stages, each containing A two-input multiplexers:

$$C_{\text{PoT}} = (W-1) \times A \times c_{\text{mux}} = (W-1) \times A \times 0.5.$$

Complexity grows *linearly* in both W and A, which is the fundamental source of PoT's hardware advantage.

FPoT shifter + adder. FPoT uses a barrel shifter with one fewer stage (shift amount $= \lfloor e/2 \rfloor$, requiring $W-2$ control bits), plus a 1-adder $\sqrt{2}$ approximation unit ($x + x{\gg}1 = 1.5x$, 6.1% error). The constant right-shift ($\gg 1$) is a hard-wired reconnection requiring zero logic gates. The 6.1% coefficient error is absorbed by *approximate-grid quantization* (Section 5.3):

$$C_{\text{FPoT}} = \underbrace{(W-2) \times A \times 0.5}_{\text{barrel shifter}} + \underbrace{1 \times A}_{\sqrt{2}\text{ adder}} = 0.5 \cdot A \cdot W.$$

Accumulator. All three modes share an identical $(W+A)$-bit accumulator adder:

$$C_{\text{ACC}} = (W+A) \times c_{\text{fa}} = W + A.$$

The INT multiplier grows as $O(WA)$ (quadratic), while both PoT and FPoT grow as $O(W + A)$ (linear). The cost advantage therefore *increases* at higher bit-widths.

Remark 1 (Assumptions and Limitations)**.** *The naïve model makes several simplifying assumptions: (1) unit cost approximation ($c_{\text{mux}} = 0.5 \times c_{\text{fa}}$); (2) no synthesis optimization; (3) no energy modeling (gate counts as area proxy only); (4) technology-node independence; (5) omitted common logic (sign negation, zero-gating), which are shared across all three modes and implemented in the accumulator control path rather than the compute unit, so their omission does not affect the CU cost ratios reported above; (6) the FPoT expression omits the even/odd output MUX and counts the $\sqrt{2}$ adder at*

Table 6: Naïve component-based cost model for INT, PoT, and FPoT compute units and full MACs (Section 4.2). Values are normalized to W4A4 INT compute unit $= 1.00\times$, using $c_{\text{mux}} = 0.5\times c_{\text{fa}}$ (full derivation in §E.1). The accumulator ($C_{\text{ACC}} = W+A$) is identical across all three modes and added to obtain the full-MAC cost.

Config	INT	PoT	FPoT	PoT/INT	FPoT/INT
Compute unit only					
W4A4	1.00	0.38	0.50	$0.38\times$	$0.50\times$
W4A8	2.00	0.75	1.00	$0.38\times$	$0.50\times$
W8A8	4.00	1.75	2.00	$0.44\times$	$0.50\times$
Full MAC (compute unit + accumulator)					
W4A4	1.50	0.88	1.00	$0.58\times$	$0.67\times$
W4A8	2.75	1.50	1.75	$0.55\times$	$0.64\times$
W8A8	5.00	2.75	3.00	$0.55\times$	$0.60\times$

activation width A, although the shifted operand it adds is wider. Assumption (6) makes the analytical FPoT cost, and with it the FPoT/INT ratio, optimistic, and §E.2 reports the synthesized 0.57–$0.59\times$ at W4. Despite these simplifications, the model correctly captures the fundamental scaling argument: multiplier cost is quadratic in bit-width while shifter + adder cost is linear, confirmed by the synthesis results.

E.2 Open-source Synthesis Results

To corroborate the structural claims of the naïve model, we synthesize the three compute units using Yosys 0.9 [Wolf and Glaser, 2013] with ABC generic-gate mapping (`abc -g gates`). Sign negation and zero-gating are implemented in the accumulator control path—consistent with Remark 1, assumption (5)—and are therefore excluded from compute-unit gate counts. Table 7 reports gate counts for all nine (config $\times$ mode) combinations.

Three observations are consistent with the analytical model. First, the structural ordering INT $>$ FPoT $>$ PoT holds for every configuration, confirming that the 1-adder $\sqrt{2}$ unit adds only moderate overhead relative to the INT multiplier. Second, the INT gate count grows faster with bit-width than FPoT: scaling from W4A4 to W8A8 (compute unit), INT grows $5.2\times$ while FPoT grows $3.8\times$, confirming that FPoT's cost advantage *widens* at higher bit-widths. Third, the full-MAC PoT/INT ratio decreases from $0.50\times$ at W4A4 to $0.37\times$ at W8A8, consistent with the quadratic-versus-linear scaling prediction.

Absolute ratios differ from Table 6 in two respects. PoT/INT at W4A4 (compute unit) measures $0.24\times$ vs. the analytical $0.38\times$: the ABC optimizer aggressively shares gates in the barrel-shifter MUX chain, making the analytical model a *conservative* estimate of PoT's advantage. FPoT/INT at W4 (compute unit) measures 0.57–$0.59\times$ against the analytical $0.50\times$. Ablative synthesis locates the difference. At W4A4 the FPoT compute unit splits into barrel shifter 10, 1-adder 26 and even/odd output MUX 8 gates, summing to the 44 of Table 7 (W4A8: 18/55/14, sum 87). The expression for C_{FPoT} in §E.1 has no term for this output MUX. Without it the unit has 36 gates at W4A4, below the 37.5 implied by the analytical $0.50\times$ (W4A8: 73 against 77), so the MUX rather than the shift path separates the measured ratio from the analytical one. The Yosys results therefore support the core structural claim of the analytical model—the INT-vs.-FPoT cost gap is real and widens with bit-width—while absolute values should be interpreted with the caveats of Remark 1. Quantitatively, the cosine similarity between the analytical and Yosys-measured cost vectors (all 18 entries of Table 6 and Table 7, normalized to W4A4 INT CU) is $\mathbf{0.988}$, confirming near-perfect agreement in structural ordering and bit-width scaling direction despite differences in absolute ratios.

E.3 MAC Energy Ratio: Published Synthesis Anchor

The energy estimates in §4.2 are anchored to the only published RTL-level ASIC synthesis comparing INT and PoT MAC units at identical conditions: Przewlocka-Rus et al. [2022] (Samsung 5 nm, 250 MHz, TT corner, 0.6 V, 25°C, 50% toggle rate). Relative to an 8×8 uniform MAC baseline, they report W4A8 uniform MAC at $2.5\times$ energy reduction and W4A8 PoT MAC at $6.0\times$. The PoT-to-INT

Table 7: Gate counts from Yosys + ABC open-source synthesis (generic-gate library, `abc -g gates`), with relative cost in parentheses normalized to W4A4 INT compute unit $= 1.00\times$ (matching Table 6). Sign negation and zero-gating reside in the accumulator control path and are excluded from compute-unit counts (Remark 1, assumption 5).

Config	INT	PoT	FPoT	PoT/INT	FPoT/INT
Compute unit only					
W4A4	75 (1.00)	18 (0.24)	44 (0.59)	$0.24\times$	$0.59\times$
W4A8	154 (2.05)	30 (0.40)	87 (1.16)	$0.19\times$	$0.57\times$
W8A8	391 (5.21)	92 (1.23)	167 (2.23)	$0.24\times$	$0.43\times$
Full MAC (compute unit + accumulator)					
W4A4	119 (1.59)	60 (0.80)	91 (1.21)	$0.50\times$	$0.77\times$
W4A8	216 (2.88)	95 (1.27)	161 (2.15)	$0.44\times$	$0.75\times$
W8A8	487 (6.49)	180 (2.40)	271 (3.61)	$0.37\times$	$0.56\times$

ratio for the *same* W4A8 precision is therefore:

$$\frac{E_{\text{PoT,W4A8}}}{E_{\text{INT,W4A8}}} = \frac{2.5}{6.0} \approx \mathbf{0.42}\times.$$

FPoT adds the $\times\sqrt{2}$ unit (one adder, active for $\sim 50\%$ of MACs). We report the FPoT/INT energy ratio as a range: 0.42–$0.46\times$, where the lower bound assumes negligible $\sqrt{2}$ switching energy and the upper bound assumes proportional contribution.

Applicability to LLMs. Although Przewlocka-Rus et al. [2022] synthesize CNN-oriented MAC units (ResNet/VGG benchmarks), the reported per-MAC energy ratio is model-agnostic: the synthesized block—a 4×8 shift-add unit vs. a 4×8 uniform multiplier—is structurally identical to the per-weight MAC in our LLM datapath. Model-level differences (sequence length, attention, layer count) affect only the aggregate MAC count, not the per-MAC energy, so the $0.42\times$ ratio transfers directly to LLM inference. Note that this anchor reports per-MAC energy of the arithmetic unit alone; total LLM inference energy additionally includes memory-access, accumulator, and activation-path contributions, the relative weight of which varies with batch size and sequence length. The per-MAC ratio therefore sets the achievable upper bound on weight-arithmetic energy savings rather than the total system-level reduction.

E.4 Per-Model Accuracy–Cost Trade-offs

The per-model figures below present accuracy–cost Pareto trade-offs across compute unit (CU) cost and full MAC cost (CU + accumulator), normalized to W4A4 Uniform INT $= 1.0\times$.

The figures reveal a consistent pattern across all seven models: (1) FPoT consistently approaches the Uniform INT accuracy at $0.50\times$ CU cost; (2) the shared accumulator compresses the advantage at full-MAC level ($0.67\times$ at W4A4); (3) as bit-widths increase, the quadratic/linear scaling gap widens, increasing FPoT's advantage; (4) all models share the same x-axis coordinates (hardware is model-independent), with variation only in accuracy (y-axis). Per-model Pareto curves for Llama-3-8B, Llama-2-7B/13B, Mistral-7B, and Qwen2.5-1.5B/7B/14B are shown in Figs. 13–19, respectively.

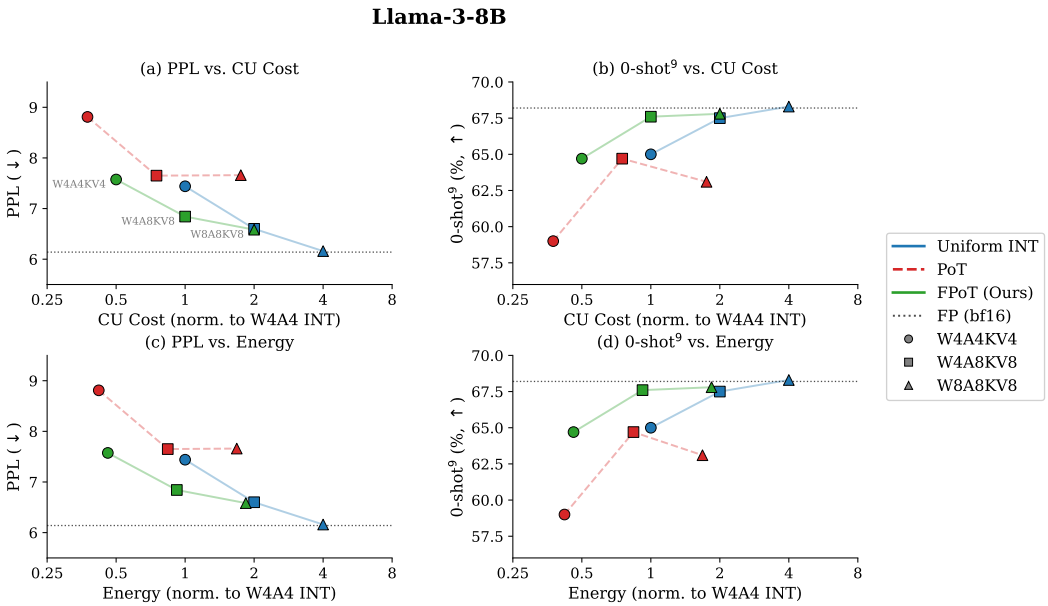


Figure 13: Accuracy–cost Pareto trade-off: Llama-3-8B.

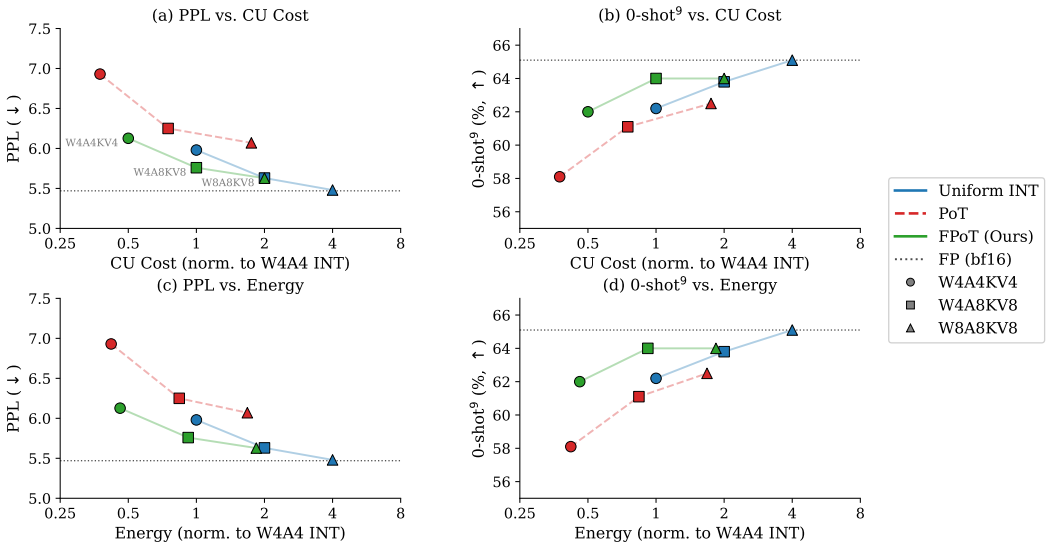


Figure 14: Accuracy–cost Pareto trade-off: Llama-2-7B.

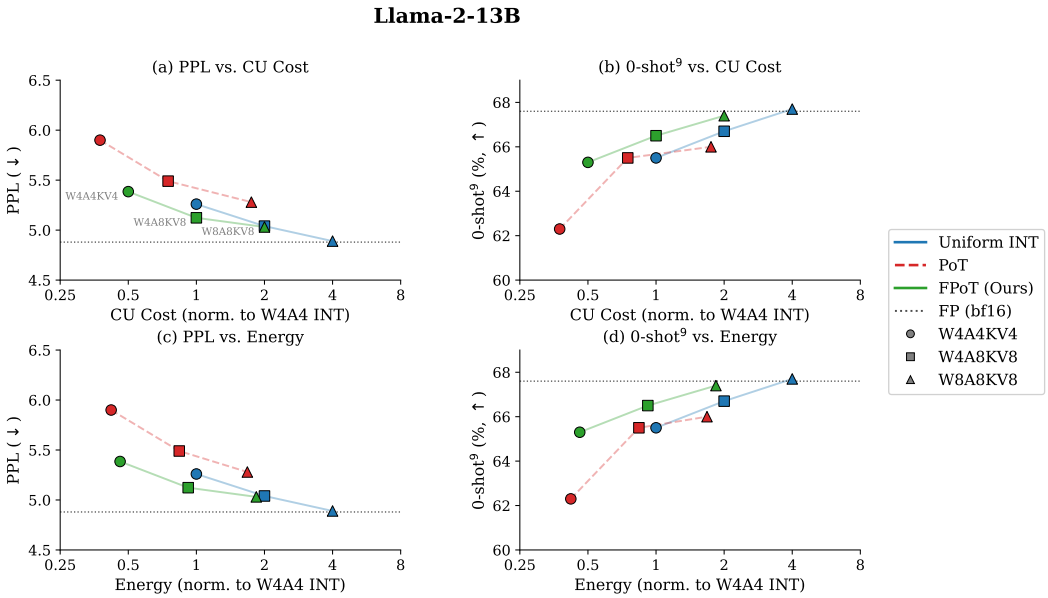


Figure 15: Accuracy–cost Pareto trade-off: Llama-2-13B.

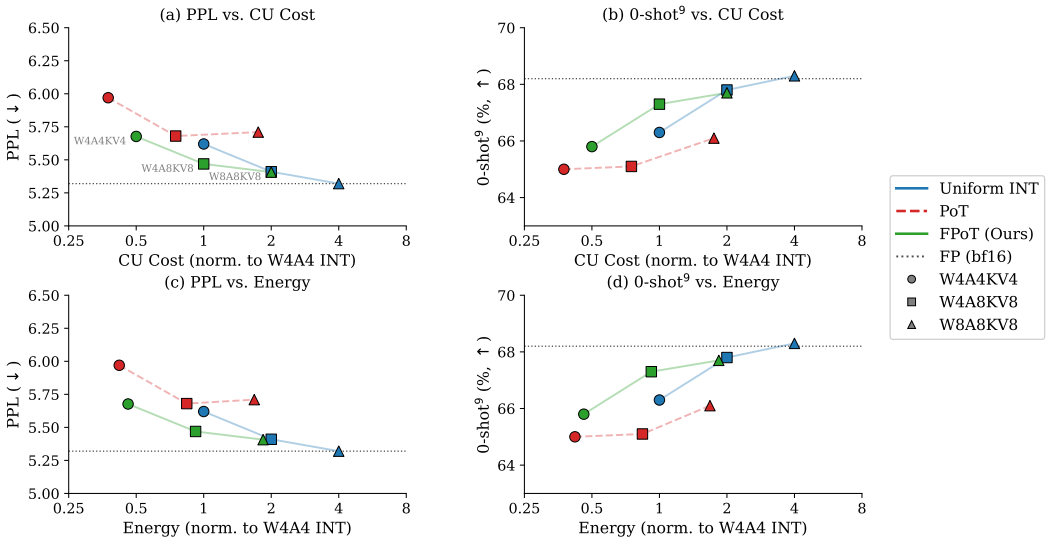


Figure 16: Accuracy–cost Pareto trade-off: Mistral-7B.

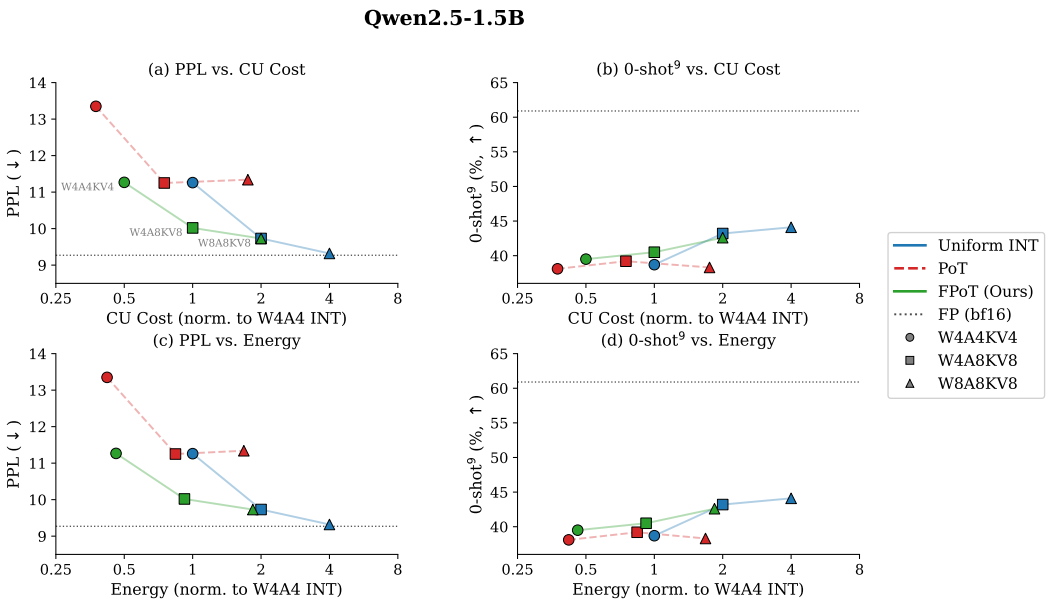


Figure 17: Accuracy–cost Pareto trade-off: Qwen2.5-1.5B.

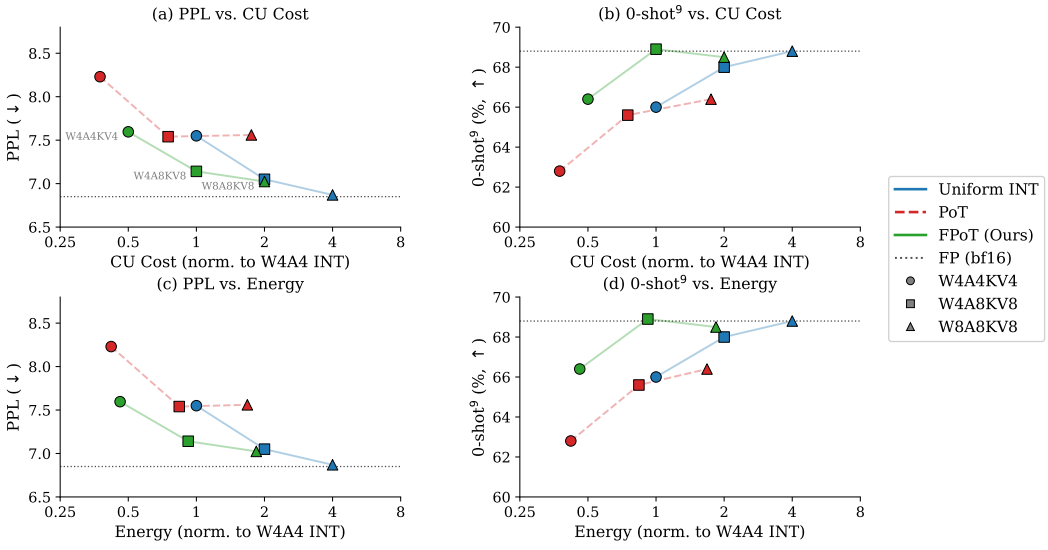


Figure 18: Accuracy–cost Pareto trade-off: Qwen2.5-7B.

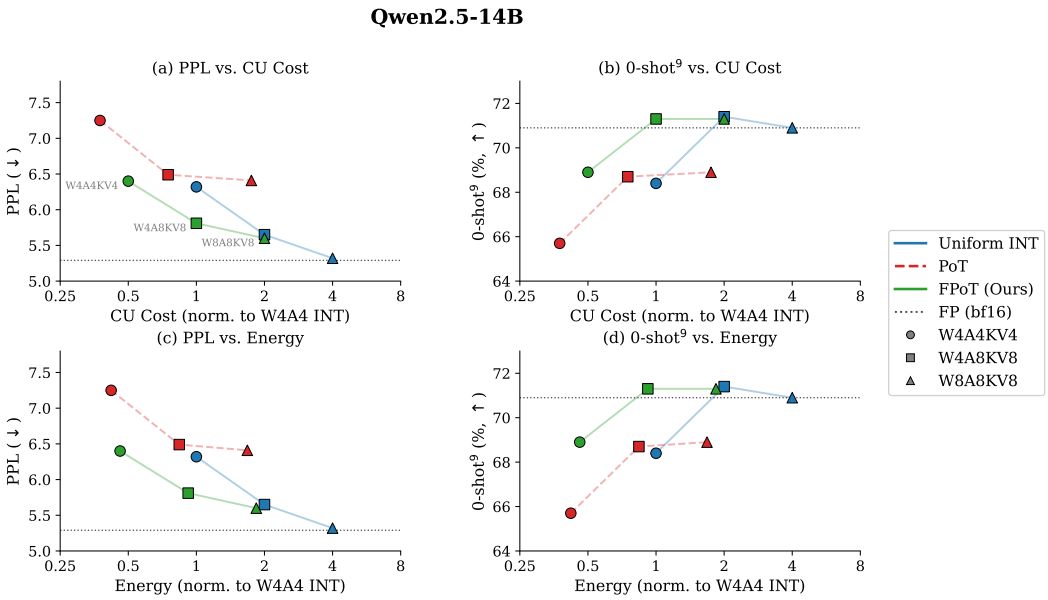


Figure 19: Accuracy–cost Pareto trade-off: Qwen2.5-14B.

F Fused Triton Kernel for FPoT Training

The FPoT fake-quantization pipeline (quantize $\rightarrow$ dequantize $\rightarrow$ STE) generates $\sim$8 intermediate tensors per weight matrix when implemented in standard PyTorch—sign, absolute value, $\log_2$ result, exponent, zero/kill masks, and dequantized output—each the same size as the weight. During gradient-checkpointed backward passes, these intermediates are regenerated, creating a memory peak that exceeds single-GPU capacity for models above $\sim$13B parameters on 80 GB GPUs.

We implement a **fused Triton kernel** that performs the entire pipeline in a single kernel launch:

1. **Quantize**: compute $|x/s|$, apply zero threshold ($2^{-0.5/k}$), compute round$(\log_2(\cdot) \times k)$, clamp exponents, determine sign and kill masks.
2. **Dequantize**: compute $s \cdot \text{sign} \cdot 2^{e/k}$.
3. **STE**: output x_{fake}; backward passes gradient through unchanged via a custom `torch.autograd.Function`.

All intermediate values reside in GPU registers and are never materialized as global-memory tensors. The kernel handles per-channel broadcast of scale, n, and m parameters natively through stride-based indexing, avoiding `expand_as().contiguous()` copies that would otherwise duplicate these tensors to full weight size.

Memory savings. On the largest single-projection weight in Qwen2.5-14B (`gate_proj`: $18{,}944 \times 5{,}120$), the fused kernel reduces peak intermediate memory from 1.07 GB to 0.19 GB—an **82% reduction**. This enables FPoT training of 14B-parameter models on a single A100-80GB with batch size 4, previously infeasible due to OOM during gradient checkpoint recomputation.

Numerical fidelity. The kernel performs all arithmetic in float32, matching the precision-critical $\log_2$ and $2^{(\cdot)}$ operations. On a full-size weight tensor, 99.6% of elements match the PyTorch reference exactly; the remaining 0.4% differ by at most one quantization level due to bf16$\rightarrow$float32 promotion order differences, with the Triton path being strictly more precise. A pure-PyTorch fallback is provided for environments without Triton.

Pseudocode. Algorithm 1 details the complete FPoT quantize/dequantize/STE pipeline.

Algorithm 1 Signed FPoT Quantization with k-Adaptive Zero Threshold

Require: Weight tensor $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$, bit budget b, resolution parameter k, clip ratio $\alpha \in (0, 1]$
Ensure: Quantized weight codes $\mathbf{M} \in \{0, \ldots, 2^{b-1}-1\}^{C_{\text{out}} \times C_{\text{in}}}$, signs $\mathbf{S} \in \{-1, 0, +1\}$, per-channel scales $\mathbf{s} \in \mathbb{R}^{C_{\text{out}}}$

// Grid parameters
$e_{\max} \leftarrow 2^{b-1} - 2$ ▷ Max exponent index (4-bit → $e_{\max}$=6)
$\tau_k \leftarrow 2^{-1/(2k)}$ ▷ k-adaptive zero threshold
(k=1: τ=0.707; k=2: τ=0.841)

// Per-channel scale with MSE clip search
for each output channel $c = 1, \ldots, C_{\text{out}}$ **do**
  $s_c \leftarrow \alpha_c^* \cdot \max_j |W_{c,j}| \;/\; 2^{e_{\max}/k}$ ▷ α_c^*: MSE-optimal clip ratio
end for

// Quantize each weight
for each weight $w = W_{c,j}$ **do**
  $\tilde{w} \leftarrow |w|/s_c$ ▷ Scale-normalized magnitude
  if $\tilde{w} < \tau_k$ **then** ▷ Below zero threshold
    $m \leftarrow 0, \quad \sigma \leftarrow 0$ ▷ Zero code (implicit pruning)
  else
    $e \leftarrow \text{clip}\big(\lfloor \log_2(\tilde{w}) \cdot k \rceil,\ 0,\ e_{\max}\big)$ ▷ FPoT exponent index
    $m \leftarrow e + 1$ ▷ Magnitude code: $m \in \{1, \ldots, 2^{b-1}-1\}$
    $\sigma \leftarrow \text{sgn}(w)$ ▷ Sign
  end if
end for

// Stored weight representation (b-bit)
`weight`$[b{-}1]$ = sign bit, `weight`$[b{-}2{:}0]$ = magnitude m

// Dequantization
$$\hat{w} = \begin{cases} 0 & \text{if } m = 0 \\ \sigma \cdot s_c \cdot 2^{(m-1)/k} & \text{if } m \geq 1 \end{cases}$$

// Hardware decomposition (k=2, inference)
For each MAC $y \mathrel{+}= x_{\text{act}} \cdot \hat{w}$:
  if $m = 0$: skip (zero-gating)
  $e \leftarrow m - 1$ ▷ Recover exponent from magnitude (subtract)
  $\text{shift} \leftarrow \lfloor e/2 \rfloor =$ `weight`$[b{-}2{:}1]$ ▷ Wire selection, 0 gates
  $\text{odd} \leftarrow e \bmod 2 =$ `weight`$[0]$ ▷ Wire, 0 gates
  $\text{shifted} \leftarrow x_{\text{act}} \ll \text{shift}$ ▷ Barrel shifter
  if even: $y \mathrel{+}= \sigma \cdot \text{shifted}$
  if odd: $y \mathrel{+}= \sigma \cdot \big(\text{shifted} + \text{shifted} \gg 1\big)$ ▷ $= 1.5 \cdot \text{shifted}$, 6.1% err vs $\sqrt{2}$, absorbed by ApproxGrid

G Additional Ablations

G.1 Calibration Sensitivity

A common concern for post-training quantization methods is sensitivity to the calibration set: both its size and domain. We evaluate FPoT on Llama-3-8B (W4A4KV4) while varying: (1) the number of calibration samples $n \in \{128, 256, 512\}$, and (2) the calibration dataset: WikiText-2 [Merity et al., 2017] (the default used throughout the paper) versus C4 [Raffel et al., 2020] (a large-scale web-crawled corpus). All other hyperparameters—OSTQuant training steps (400), learning rate, GPTQ settings—remain fixed.

Table 8: Calibration sensitivity analysis on Llama-3-8B (W4A4KV4, FPoT, 400 steps). Rows vary the calibration dataset and number of calibration samples (n). The default configuration used throughout the paper is WikiText-2 with n=128.

Cal. Dataset	n	PPL↓	0-shot[9]↑ (%)	ΔPPL
WikiText-2	128	7.559	65.07	—
	256	7.589	64.68	+0.030
	512	7.525	64.80	−0.034
C4 [Raffel et al., 2020]	128	7.617	64.91	+0.058
	256	7.660	65.01	+0.101
	512	7.647	64.88	+0.088

FPoT is robust to calibration set size. Within either dataset, PPL varies by at most 0.065 (7.525–7.589 for WikiText-2; 7.617–7.660 for C4) across a $4\times$ range of sample counts. The 0-shot[9] accuracy is similarly stable, with a maximum spread of 0.4%p. This indicates that n=128 samples are sufficient for FPoT calibration; increasing to 256 or 512 yields no systematic improvement, consistent with prior findings for GPTQ-based methods [Frantar et al., 2023].

FPoT is robust to calibration domain. Switching from WikiText-2 to C4 increases PPL by $+0.06$–$+0.10$, a minor shift relative to the method's $+0.12$ total overhead. Downstream accuracy is virtually unchanged (64.9–65.0% for C4 vs. 64.7–65.1% for WikiText-2). The modest cross-domain degradation suggests that FPoT's log-scale grid structure is inherently less sensitive to calibration distribution than fine-grained uniform quantization methods, since the quantization boundaries (powers of two) are data-independent—only the per-channel scale s adapts to the calibration data.

Practical recommendation. Our default setting (n=128, WikiText-2) is near-optimal across all configurations tested, and practitioners may safely use alternative calibration sources without re-tuning.

We evaluate design choices in the FPoT quantization scheme: the zero-threshold parameter, rounding mode, and weight-only vs. full PoT quantization. All experiments use Llama-3-8B (W4A4KV4, FPoT, 400 OSTQuant training steps).

G.2 Weight Grid-Alignment

Table 9 reports PPL and 0-shot[9] across $G \in \{4, 8, 16, 32\}$ phase-shift candidates on Llama-3-8B (W4A4KV4, 400 steps).

Table 9: Weight grid-alignment ablation (Llama-3-8B, W4A4KV4, 400 steps, single seed). G: number of phase-shift candidates (Eq. (4)); G=— denotes the plain FPoT baseline. At k=1, grid-alignment reduces PPL by up to -0.051; at k=2 the finer grid already self-corrects, so $|\Delta\text{PPL}| \leq 0.034$. 0-shot[9] values are in %.

	k=1		k=2	
G	PPL↓	0-shot[9]↑	PPL↓	0-shot[9]↑
—	8.806	58.99	7.563	64.28
4	8.755	59.09	7.545	64.49
8	8.771	60.04	7.597	64.45
16	8.771	60.08	7.560	64.71
32	8.801	59.69	7.568	63.55

G.3 Weight-Only vs. Full PoT Quantization

Table 10 shows that applying PoT/FPoT to both weights and activations is catastrophic (PPL 16.9–114.1 at k=1; even FPoT$_{k=2}$ reaches only 8.21–52.49), while weight-only PoT/FPoT remains usable. The dominant bottleneck is the coarse logarithmic grid applied to post-softmax activations, not activation quantization per se.

Table 10: Weight-only vs. Full PoT/FPoT on Llama-3-8B (all 7 projections, single seed). k=1 is standard PoT and k=2 is FPoT. W-only: PoT/FPoT weight + uniform INT activation. Full: PoT/FPoT applied to both weight and activation. 0-shot[9] values are in %.

Config	k	Scope	PPL↓	0-shot[9]↑
W4A4KV4	Uniform (non-PoT)		7.44	65.0
	k=1	Full	16.91	44.6
	k=1	W-only	8.81	59.0
	k=2	Full	8.21	62.5
	k=2	**W-only (Ours)**	**7.56**	**64.3**
W4A8KV8	Uniform (non-PoT)		6.60	67.5
	k=1	Full	114.1	42.7
	k=1	W-only	7.65	64.7
	k=2	Full	52.49	53.8
	k=2	**W-only (Ours)**	**6.80**	**67.2**
W8A8KV8	Uniform (non-PoT)		6.16	68.3
	k=1	Full	37.07	52.3
	k=1	W-only	7.66	63.1
	k=2	Full	27.91	57.4
	k=2	**W-only (Ours)**	**6.54**	**66.7**

G.4 k-Adaptive Zero-Threshold Sensitivity

In FPoT quantization, values with $|w/s| < \tau$ are mapped to zero rather than to the nearest PoT level. We parameterize $\tau = 2^{\alpha/k}$ where α controls the aggressiveness of zeroing. The default $\alpha = -0.5$ places the threshold at the log-space midpoint between zero and the smallest PoT level ($s \cdot 2^0 = s$), which is the theoretically optimal boundary under a uniform-in-log prior.

Table 11: Zero-threshold sensitivity analysis on Llama-3-8B (W4A4KV4, FPoT, 400 steps). The threshold $\tau = 2^{\alpha/k}$ determines the boundary below which $|w/s|$ is mapped to zero. The default α=-0.5 corresponds to the log-space midpoint between zero and the smallest PoT level.

α	τ (=$2^{\alpha/2}$)	PPL↓	0-shot[9]↑ (%)
$\mathbf{-0.5}$	$\mathbf{0.841}$	$\mathbf{7.560}$	$\mathbf{64.71}$
-1	0.707	7.573	64.99
-2	0.500	7.565	64.99
-3	0.354	7.553	65.06
-4	0.250	7.545	64.72
-6	0.125	7.557	64.54
-100	$\approx$0	7.540	64.78

Table 11 sweeps α from -0.5 (aggressive zeroing, $\tau = 0.841$) to -100 (effectively no zeroing, $\tau \approx 0$). Across this entire range, PPL varies by only 0.033 (7.540–7.573) and 0-shot[9] by 0.5%p (64.5–65.1%). The negligible sensitivity indicates that the zero-threshold is **not a critical hyperparameter** for FPoT: state-of-the-art LLM PTQ preprocessing and GPTQ calibration effectively compensate for the threshold choice. Practitioners may safely use the default $\alpha = -0.5$ without tuning.

G.5 Rounding Mode

The standard FPoT rounding uses nearest-neighbor in log-space: $e = \text{round}(\log_2(|w/s|) \cdot k)$, which places decision boundaries at the *geometric mean* of adjacent PoT levels. An alternative is linear-space nearest-neighbor, which places boundaries at the *arithmetic mean*—potentially better aligned with MSE-based objectives like GPTQ, since it minimizes absolute (rather than relative) quantization error per element.

Table 12: Rounding mode comparison on Llama-3-8B (W4A4KV4, FPoT, 400 steps). **Log-space** rounds to the nearest PoT level measured by relative distance (geometric mean boundary). **Linear-space** rounds by absolute distance (arithmetic mean boundary).

Rounding mode	PPL↓	0-shot[9]↑ (%)
Log-space NN (default)	7.560	64.71
Linear-space NN	7.559	64.81

Table 12 shows that the two rounding modes produce **virtually identical results** (ΔPPL $= -0.001$, Δ0-shot^9 $= +0.1$%p). This is expected: for FPoT, adjacent levels differ by a factor of $\sqrt{2} \approx 1.414$, so the geometric and arithmetic mean boundaries differ by only $\sim$1.5%—well within the noise floor of the calibration process. The default log-space rounding is retained for its computational simplicity (one `round` vs. `floor` + two `exp2` + comparison).

G.6 Preprocessing Ablation

Table 13 ablates the type and quality of the preprocessing transform under FPoT weight quantization on Llama-3-8B W4A4KV4. Configurations vary along two orthogonal axes: rotation quality (none $\rightarrow$ fixed Hadamard $\rightarrow$ learned) and the presence of per-channel scale.

Table 13: Preprocessing component ablation within OSTQuant under FPoT weight quantization on Llama-3-8B W4A4KV4 (400 steps, single seed; cf. Table 1 3-seed mean $7.60_{\pm.02}$ for the full-pipeline entry). Activations and KV-cache use uniform INT4 throughout.

Preprocessing	PPL↓	0-shot[9]↑ (%)
No transform	181.3	37.56
Scale only	9.78	58.79
Fixed Hadamard	8.30	62.64
Learned rotation	7.78	63.30
Learned rotation + scale	**7.56**	**64.28**

Both rotation and scale are necessary. Without any transform, FPoT yields catastrophic PPL $=$ 181.3. Fixed Hadamard rotation alone (8.30) or learned rotation alone (7.78) provide substantial recovery, while scale-only achieves 9.78. Only the full combination (learned rotation + scale) reaches 7.56, within $+0.12$ of the non-PoT baseline, because rotation suppresses position-level outliers on the logarithmic grid while scale normalizes per-channel amplitude. More broadly, this supports the design criterion satisfied by state-of-the-art LLM PTQ methods such as OSTQuant and DuQuant.

H Per-Task Downstream Results

This appendix reports individual zero-shot task scores for all models and bit-width configurations in Table 1. For tasks providing both `acc` and `acc_norm`, we report `acc_norm` (ARC-C, ARC-E, HellaSwag, PIQA, OpenBookQA); all others report `acc`. The "AVG" column is the unweighted mean across all nine tasks, matching the 0-shot[9] metric in the main paper. All scores are in %. Three weight quantization methods are compared: **Uniform** (uniform INT baseline), **PoT** (weight-only), and **FPoT (Ours)**, with the unquantized FP (bf16) reference shown above each model's quantized results. Tables 14–20 report per-task results for each model across the W4A4KV4, W4A8KV8, and W8A8KV8 configurations. **FPoT (Ours) rows are bolded throughout.**

Observations. Across all seven models and three bit-width configurations, FPoT with 1-adder approximate-grid quantization maintains competitive per-task performance relative to both the FP (bf16) reference and the Uniform INT baseline. Key patterns:

- **Per-task Δ within ± 3%p**: On most tasks, the difference between uniform INT and FPoT is within $\pm$2–3%p, consistent with the aggregate 0-shot[9] results in Table 1.
- **WinoGrande** consistently *improves* under FPoT at W4A4KV4 for five of seven models, suggesting a mild regularization benefit from the logarithmic weight grid.

Table 14: Per-task results: **Llama-3-8B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		53.1	77.9	79.2	73.4	81.0	81.3	45.0	75.7	47.1	68.2
W4A4KV4	Uniform	48.9	75.0	75.9	68.2	78.1	77.0	42.0	73.5	46.3	65.0
	PoT	39.7	63.6	71.1	67.2	74.3	67.2	37.2	68.0	42.6	59.0
	FPoT (Ours)	**45.6**	**73.4**	**74.7**	**69.2**	**76.8**	**78.0**	**41.6**	**73.2**	**45.1**	**64.2**
W4A8KV8	Uniform	53.3	78.0	77.8	74.0	79.6	78.5	43.6	75.3	47.5	67.5
	PoT	47.2	72.6	74.8	72.1	78.0	78.1	42.6	71.8	45.5	64.7
	FPoT (Ours)	**51.8**	**78.0**	**77.0**	**72.8**	**79.2**	**79.5**	**43.8**	**75.2**	**46.6**	**67.1**
W8A8KV8	Uniform	53.7	78.4	79.1	73.3	81.0	81.6	45.6	75.4	46.7	68.3
	PoT	43.9	71.6	74.6	72.5	76.2	68.3	42.4	72.6	45.5	63.1
	FPoT (Ours)	**51.5**	**76.9**	**78.2**	**74.5**	**80.7**	**80.6**	**44.4**	**75.0**	**46.0**	**67.5**

Table 15: Per-task results: **Llama-2-7B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		45.9	74.5	75.9	68.8	78.7	78.0	44.2	73.6	46.0	65.1
W4A4KV4	Uniform	42.8	69.8	73.1	66.9	77.4	74.7	41.2	70.4	44.0	62.2
	PoT	38.7	65.6	69.7	62.8	75.4	68.8	37.2	63.8	41.1	58.1
	FPoT (Ours)	**41.6**	**70.3**	**71.9**	**67.3**	**76.7**	**73.8**	**40.6**	**72.4**	**44.0**	**62.1**
W4A8KV8	Uniform	45.3	74.0	74.7	68.4	78.0	75.3	42.6	71.6	44.4	63.8
	PoT	42.3	68.2	72.3	66.5	76.9	72.3	41.2	66.7	43.6	61.1
	FPoT (Ours)	**42.8**	**72.1**	**73.8**	**68.9**	**78.5**	**76.6**	**40.6**	**73.6**	**44.6**	**63.5**
W8A8KV8	Uniform	45.4	74.6	75.9	69.7	79.1	77.9	44.0	73.2	46.0	65.1
	PoT	43.0	70.6	74.1	67.6	78.4	73.0	43.8	67.4	44.5	62.5
	FPoT (Ours)	**44.2**	**72.4**	**75.0**	**68.6**	**77.8**	**75.9**	**43.2**	**73.4**	**45.0**	**64.0**

- **PoT degradation is broad**: Standard PoT falls below uniform INT in 176 of the 189 task–configuration cells, by as much as 24.2%p (BoolQ), so the coarse grid affects nearly every downstream task rather than a few.
- **Qwen2.5 and Mistral-7B: FPoT matches or exceeds uniform**: At W4A4KV4, FPoT AVG meets or surpasses the uniform INT baseline for all three Qwen2.5 models ($+0.7$–$+1.2$%p) and Mistral-7B ($+0.1$%p), while the Llama family shows a small deficit (-0.1 to -0.8%p). FPoT overhead is thus model-family dependent rather than universally negative.
- **LAMBADA: contrasting patterns across models**: FPoT consistently outperforms Uniform on Llama-2 and Mistral-7B by $+0.2$–$+2.0$%p across all bit-widths, suggesting a task-specific affinity between FPoT's sparse weight representation and long-range language modeling. By contrast, all quantized methods collapse to $\approx 0\%$ for Qwen2.5-1.5B (FP reference: 61.9%).
- **FP gap shrinks with bit-width**: At W8A8KV8, FPoT AVG is within ~ 1%p of the FP reference for six of seven models (Qwen2.5-1.5B is an exception at -18.7%p, reflecting its small-model sensitivity to aggressive quantization). The W4A4KV4 gap is largest across all models.

Table 16: Per-task results: **Llama-2-13B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		49.1	77.5	79.4	71.7	80.6	80.7	45.6	76.7	47.2	67.6
W4A4KV4	Uniform	45.7	73.6	77.0	69.6	79.1	77.8	45.0	75.4	46.8	65.5
	PoT	44.5	70.4	74.8	67.8	77.0	75.7	40.8	67.8	42.1	62.3
	FPoT (Ours)	**47.2**	**73.9**	**75.9**	**68.7**	**78.0**	**78.8**	**43.6**	**75.7**	**44.8**	**65.2**
W4A8KV8	Uniform	48.1	76.1	78.3	72.5	79.9	79.4	43.2	76.6	46.6	66.7
	PoT	49.6	75.8	76.8	69.6	78.8	79.5	42.0	72.1	45.1	65.5
	FPoT (Ours)	**48.6**	**76.6**	**77.2**	**71.4**	**79.6**	**78.7**	**42.4**	**77.4**	**46.9**	**66.5**
W8A8KV8	Uniform	49.3	77.4	79.4	72.1	80.5	80.6	46.0	76.6	47.3	67.7
	PoT	48.6	75.6	77.7	69.1	79.2	79.7	45.0	73.3	45.9	66.0
	FPoT (Ours)	**49.0**	**77.6**	**78.4**	**71.4**	**80.3**	**80.6**	**46.4**	**77.1**	**46.7**	**67.5**

Table 17: Per-task results: **Qwen2.5-1.5B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		44.9	71.6	67.8	63.5	76.0	72.8	40.4	61.9	49.0	60.9
W4A4KV4	Uniform	30.9	36.5	41.0	58.0	62.0	51.0	33.4	0.0	35.9	38.7
	PoT	31.3	34.9	38.8	54.2	61.4	53.1	34.2	0.0	35.2	38.1
	FPoT (Ours)	**32.5**	**37.5**	**40.3**	**59.0**	**62.7**	**56.3**	**35.0**	**0.0**	**35.6**	**39.9**
W4A8KV8	Uniform	33.5	39.3	43.7	62.8	64.5	70.9	37.0	0.0	36.9	43.2
	PoT	33.0	38.2	40.6	61.1	61.9	46.7	34.8	0.0	36.2	39.2
	FPoT (Ours)	**33.4**	**39.1**	**43.1**	**62.6**	**64.2**	**56.3**	**36.4**	**0.0**	**36.8**	**41.3**
W8A8KV8	Uniform	35.1	40.8	45.7	63.5	65.2	71.0	36.4	0.8	38.0	44.1
	PoT	31.4	34.7	39.3	59.8	61.8	49.3	33.0	0.0	35.2	38.3
	FPoT (Ours)	**34.6**	**40.5**	**44.0**	**62.4**	**64.6**	**61.5**	**34.8**	**0.1**	**37.4**	**42.2**

Table 18: Per-task results: **Qwen2.5-7B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		51.4	77.7	79.0	73.2	80.0	84.7	47.0	71.6	54.8	68.8
W4A4KV4	Uniform	49.2	75.3	76.1	69.9	79.1	81.5	44.4	68.4	50.3	66.0
	PoT	46.3	70.9	74.6	64.8	77.2	79.1	41.4	61.6	49.4	62.8
	FPoT (Ours)	**52.3**	**79.6**	**75.3**	**67.5**	**78.4**	**82.3**	**43.8**	**69.2**	**51.9**	**66.7**
W4A8KV8	Uniform	52.4	78.3	78.2	71.6	79.4	83.9	44.8	71.8	51.8	68.0
	PoT	49.4	73.9	76.6	70.9	79.4	81.6	43.2	64.7	50.8	65.6
	FPoT (Ours)	**55.8**	**81.2**	**77.7**	**72.3**	**80.0**	**85.8**	**44.4**	**71.5**	**54.4**	**69.2**
W8A8KV8	Uniform	51.8	77.8	78.8	72.6	79.9	84.5	47.0	71.3	55.0	68.8
	PoT	50.3	75.4	76.8	69.1	79.1	83.8	45.2	66.4	51.1	66.4
	FPoT (Ours)	**53.6**	**79.5**	**78.3**	**71.3**	**79.9**	**83.8**	**44.8**	**71.8**	**53.6**	**68.5**

Table 19: Per-task results: **Qwen2.5-14B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		59.0	79.3	82.9	74.7	82.1	85.4	44.8	74.1	55.5	70.9
W4A4KV4	Uniform	55.2	80.5	80.0	72.1	80.3	83.9	42.0	71.3	50.0	68.4
	PoT	54.5	77.6	77.1	66.6	78.6	80.6	43.0	65.3	47.8	65.7
	FPoT (Ours)	**54.7**	**81.3**	**79.7**	**73.2**	**79.7**	**84.6**	**43.6**	**73.6**	**52.2**	**69.2**
W4A8KV8	Uniform	60.5	81.0	82.5	75.5	82.1	85.4	45.0	75.4	55.2	71.4
	PoT	57.4	82.0	80.3	73.0	80.9	82.6	42.0	68.0	51.8	68.7
	FPoT (Ours)	**59.6**	**83.7**	**82.0**	**78.0**	**82.4**	**87.7**	**45.2**	**75.3**	**54.3**	**72.0**
W8A8KV8	Uniform	58.1	78.8	82.9	75.5	82.0	85.0	45.8	74.1	55.7	70.9
	PoT	55.4	79.7	80.9	73.8	80.9	84.6	43.8	70.0	50.9	68.9
	FPoT (Ours)	**60.3**	**82.9**	**82.3**	**76.8**	**82.7**	**86.8**	**46.0**	**74.7**	**55.0**	**71.9**

Table 20: Per-task results: **Mistral-7B**. Single calibration seed (seed 42); FP (bf16) is the unquantized reference; **FPoT (Ours)** rows are bolded. All scores in %.

Config	Method	ARC-C	ARC-E	Hella	Wino	PIQA	BoolQ	OBQA	Lam	SIQA	AVG
FP (bf16) reference		52.2	78.3	80.5	73.6	82.3	82.1	44.2	75.1	45.8	68.2
W4A4KV4	Uniform	48.7	75.8	78.7	69.9	80.4	82.4	43.2	73.4	44.7	66.3
	PoT	47.3	74.7	77.4	68.7	80.9	79.9	41.6	70.5	44.3	65.0
	FPoT (Ours)	**48.9**	**77.1**	**77.6**	**70.5**	**80.7**	**81.0**	**42.0**	**75.0**	**44.7**	**66.4**
W4A8KV8	Uniform	50.7	78.2	79.7	73.0	81.7	82.3	44.2	74.6	45.4	67.8
	PoT	47.3	74.5	78.1	71.1	80.5	76.7	41.4	71.7	44.9	65.1
	FPoT (Ours)	**50.4**	**78.1**	**78.6**	**73.0**	**81.0**	**80.9**	**41.6**	**76.3**	**45.3**	**67.3**
W8A8KV8	Uniform	52.6	78.5	80.4	73.6	82.0	82.3	44.0	75.0	46.0	68.3
	PoT	48.3	75.3	79.2	71.1	80.7	79.1	44.2	71.5	45.8	66.1
	FPoT (Ours)	**51.0**	**78.1**	**79.3**	**72.5**	**81.8**	**83.4**	**43.2**	**76.3**	**45.2**	**67.9**

I Comparison with LogART

LogART [Xu et al., 2026] is a concurrent logarithmic quantization method that also targets multiplier-free inference via power-of-two weight representations. Both methods share the same hardware motivation—replacing integer multipliers with barrel shifters—but differ in scope and design:

- **LogART**: Dynamic base assignment (per-channel base-2 or base-$\sqrt{2}$), learned rounding (500 iterations/layer), asymmetric quantization. *Weight-only*; no activation or KV-cache quantization is reported for LLMs.
- **FPoT (Ours)**: Fixed base-$\sqrt{2}$ grid, 1-adder approximate-grid quantization, state-of-the-art LLM PTQ preprocessing. *Full W/A/KV quantization* within a unified framework.

Important caveat: LogART's official implementation excludes activation quantization. For the W4A4KV4 rows below, we apply our per-token dynamic INT4 activation quantizer post-hoc to LogART's weight-quantized model without PTQ preprocessing. These results analyze the interaction between activation quantization and distribution redistribution, not a direct LogART benchmark.

Table 21 compares the two methods under weight-only (W4A16) and full (W4A4KV4) quantization. LogART is reproduced using the official open-source code[1] with the same calibration setup as our experiments.

Table 21: Comparison with LogART under weight-only (W4A16) and full (W4A4KV4) quantization. All methods are multiplier-free (PoT-based). FPoT W4A4KV4 entries are copied from Table 1 (mean over three calibration seeds); FPoT W4A16 and all LogART entries use a single calibration seed. PPL: WikiText-2 (↓). 0-shot[9]: unweighted mean across nine benchmarks (↑, %).

		W4A16		**W4A4KV4**[†]	
Model	**Method**	PPL↓	0-shot[9]↑	PPL↓	0-shot[9]↑
Llama-3-8B	LogART	6.516	68.8	3903	34.4
	FPoT (Ours)	6.862	66.6	**7.60**	**64.8**
Llama-2-7B	LogART	5.617	66.8	∞	33.9
	FPoT (Ours)	5.861	63.5	**6.11**	**61.9**

[†] Post-hoc INT4 activation quantization applied to LogART's W4 model without PTQ preprocessing. This is *not* LogART's official configuration. PPL $= \infty$ indicates numerical overflow (NaN loss).

In the weight-only setting (W4A16), LogART achieves lower PPL and higher 0-shot accuracy than FPoT on both models (-0.24 PPL / $+3.3$%p 0-shot on Llama-2-7B; -0.35 PPL / $+2.2$%p 0-shot on Llama-3-8B), likely due to (1) dynamic per-channel base allocation and (2) learned rounding that directly optimizes the Hessian-weighted reconstruction loss.

However, when INT4 activation quantization is applied, LogART's weight-quantized model degrades catastrophically (PPL > 3900 or ∞; 0-shot[9] drops to $\sim$34%, near random), while FPoT maintains stable accuracy (PPL ≤ 7.60, 0-shot[9] ≥ 61.9%). This confirms that **distribution redistribution via state-of-the-art LLM PTQ preprocessing is a prerequisite for stable joint W4A4KV4 quantization**. Without Hadamard rotation, channel-concentrated activation outliers dominate the per-token scale, effectively quantizing the majority of activations to zero. FPoT's compatibility with state-of-the-art LLM PTQ preprocessing resolves this at the source by redistributing outliers prior to quantization, enabling stable W4A4KV4 deployment that weight-only methods do not support.

J Cross-Framework Portability

A natural question is whether FPoT's low overhead is specific to OSTQuant's learned rotation–scaling pipeline or transfers to other post-training quantization (PTQ) frameworks. To answer this, we integrate FPoT weight quantization into DuQuant [Lin et al., 2024a], a recent PTQ method that uses

[1] LogART settings: `log_dynamic` + `scale_search` + `learn_rounding` (500 iters, lr $= 0.05$), asymmetric, per-channel, no Z-Fold or OPTQ init. Calibration: WikiText-2, 128 samples, seqlen 2048. The LogART runs use exact $\sqrt{2}$, without its hardware approximation, and their 0-shot average is over LogART's task set, which has RTE in place of Social IQA.

a *dual-transform* strategy—iteratively applying learned block-diagonal rotations and zigzag channel permutations to smooth both weight and activation outliers before quantization.

Integration. We replace DuQuant's round-to-nearest (RTN) weight quantizer with a column-wise weight calibration backend that supports both uniform INT and FPoT grids. Activations and the KV-cache remain on the uniform INT grid in all configurations; only the weight grid is swapped. We use symmetric per-channel weight quantization, no learnable weight clipping (epochs = 0), and a single calibration seed (WikiText-2, 128 samples, seqlen 2048). The weight calibration backend performs its own MSE-based clip-ratio search per channel, independent of DuQuant's smooth/rotation transforms.

Results. Table 22 compares OSTQuant (our main pipeline) and DuQuant+GPTQ under identical evaluation (WikiText-2 perplexity and nine zero-shot benchmarks). Several observations are worth highlighting:

- **FPoT overhead is small and framework-invariant.** Across all completed model–config pairs, replacing uniform INT weights with FPoT increases perplexity by $+0.05$–0.34 in the DuQuant pipeline, comparable to the $+0.08$–0.42 range observed with OSTQuant (Table 1). The overhead is consistently smallest at W4A4KV4, where the quantization grid is already coarse and the additional granularity of the FPoT half-octave grid compensates well.
- **DuQuant matches OSTQuant on absolute quality.** For Llama-2-7B, the two frameworks produce nearly identical perplexity under both uniform and FPoT grids (within ± 0.15 PPL across all bit-widths). For Llama-3-8B the gap is slightly larger (≈ 0.3 PPL at W4A4), reflecting differences in the preprocessing strategy rather than the weight grid.
- **Zero-shot accuracy tracks perplexity.** The nine-task mean accuracy drops by at most 1.1%p when switching from uniform to FPoT—within the noise margin of single-seed runs—confirming that the perplexity overhead translates to negligible downstream impact.

These results demonstrate that FPoT's near-lossless multiplier-free property is not tied to a specific PTQ framework: any pipeline that performs channel-level weight calibration (e.g. column-wise MSE search or OSTQuant's learned scaling) can adopt FPoT weights with minimal additional quality loss.

Table 22: Cross-framework comparison of FPoT weight quantization; only the weight grid differs across methods (Uniform INT vs. FPoT), with activations and KV-cache using uniform INT throughout. **OSTQuant** (ours): 3-seed mean; see Table 1. **DuQuant**: single calibration seed; column-wise weight calibration replaces the default RTN quantizer (see Appendix J). PPL: WikiText-2 perplexity (↓). 0-shot[9]: mean zero-shot accuracy across nine benchmarks (↑, %).

			W4A4KV4		W4A8KV8		W8A8KV8	
Model	Method	Weight	PPL↓	0-shot[9]↑	PPL↓	0-shot[9]↑	PPL↓	0-shot[9]↑
Llama-3-8B	OSTQuant	Uniform	7.36	65.2	6.60	67.5	6.16	68.2
		FPoT	7.60	64.8	6.85	67.3	6.58	67.7
	DuQuant	Uniform	7.67	62.9	6.56	67.3	6.17	68.0
		FPoT	7.87	63.1	6.68	66.6	6.51	66.9
Llama-2-7B	OSTQuant	Uniform	5.97	62.1	5.63	63.8	5.48	65.1
		FPoT	6.11	61.9	5.75	63.8	5.65	64.2
	DuQuant	Uniform	6.08	61.7	5.62	64.3	5.49	65.0
		FPoT	6.19	61.6	5.67	63.3	5.60	64.4
Llama-2-13B	OSTQuant	Uniform	5.26	65.6	5.03	67.1	4.89	67.7
		FPoT	5.37	65.2	5.11	66.6	5.03	67.2
	DuQuant	Uniform	5.30	65.5	5.00	66.6	4.90	67.4
		FPoT	5.35	65.4	5.05	67.1	4.99	66.8

K Preprocessing × Quantizer Factorial

FPoT is always paired with learned preprocessing, so its gain over PoT could come from the grid itself or from preprocessing trained for that grid. Table 23 separates the two on Llama-3-8B W4A4KV4. The OSTQuant rotation and scales are trained for one of three grids (rows), and GPTQ then quantizes to one of three grids (columns), with three seeds per cell. The weight quantizer dominates. Column means span 1.51 PPL and 5.51%p accuracy, while row means span 0.15 PPL and 0.97%p. The order INT, FPoT, PoT (best to worst) holds in every row for both metrics, so FPoT keeps its advantage over PoT whichever preprocessing it is paired with. On the diagonal, which holds the runs of Table 1, FPoT recovers 83.4% of the PoT→INT PPL gap and 92.3% of the accuracy gap.

Table 23: Preprocessing × weight-quantizer factorial on Llama-3-8B W4A4KV4 (mean±std over 3 seeds). Rows give the grid that the OSTQuant rotation and scales were trained for, and columns give the grid that GPTQ quantizes to. PPL is WikiText-2 perplexity (↓) and 0-shot[9] is the mean zero-shot accuracy over nine tasks (↑, %). **Bold** marks the matched pairs (diagonal), which are the runs of Table 1. The last column averages each row, and the mean row of each block averages each column.

		Weight quantizer			
Metric	Preprocessing target	Uniform	PoT	FPoT (Ours)	mean
PPL↓	Uniform	**7.358** ±.074	9.160 ±.126	7.851 ±.039	8.123
	PoT	7.493 ±.022	**8.786** ±.037	7.833 ±.042	8.037
	FPoT (Ours)	7.437 ±.023	8.886 ±.061	**7.596** ±.020	7.973
	mean	7.430	8.944	7.760	
0-shot[9]↑	Uniform	**65.22** ±.37	58.69 ±.60	63.16 ±.44	62.35
	PoT	64.67 ±.31	**59.55** ±.58	63.39 ±.44	62.54
	FPoT (Ours)	65.03 ±.38	60.15 ±.91	**64.78** ±.55	63.32
	mean	64.97	59.46	63.78	

L Generation and Instruction Following

Quantization error can accumulate over autoregressive decoding, which perplexity and zero-shot accuracy do not exercise. Table 24 evaluates two instruction-tuned models at W4A4KV4 on math, code, instruction following, safety and long-context retrieval. On math and code, standard PoT trails uniform INT by about 15 to 25 points. On Llama-3.1-8B-Instruct its HumanEval score falls from 66.5 (bf16) to 27.6, against 52.4 for uniform INT, and FPoT restores it to 47.8, recovering 81% of the PoT→INT gap. FPoT recovers 77–88% of that gap on Qwen2.5-7B-Instruct math and code. On Llama it also closes the long-context gap (79.4 against INT 79.6 and PoT 61.8) and 60% of the safety regression PoT introduces. Safety on Qwen is saturated (bf16 attack-success rate 0.00%), and on Qwen long-context retrieval FPoT does not improve over PoT.

Protocol. Table 24 applies the preprocessing and GPTQ pipeline of the main results to Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, with three calibration seeds (0, 42, 123) per quantized method. GSM8K uses the lm-eval-harness `gsm8k` task (5-shot, flexible-extract exact match, no chat template). IFEval reports prompt-level strict accuracy under the model's chat template. HumanEval prompts the model through its chat template, extracts the code block from the reply, and reports pass@1 against the unit tests. The 520 AdvBench prompts are also issued through the chat template, Llama-Guard-3-8B judges each reply, and the reported value is the fraction judged unsafe (attack-success rate). LongBench uses the English passage-retrieval task (40 samples) with inputs truncated to 10,000 tokens for Qwen and 8,000 for Llama.

Table 24: Generation and instruction following at W4A4KV4 (mean±std over 3 seeds; FP (bf16) is the unquantized model). GSM8K exact match, HumanEval pass@1, IFEval prompt-level strict accuracy, AdvBench attack-success rate (↓) judged by Llama-Guard-3-8B, and LongBench passage retrieval. **FPoT (Ours)** rows are bolded.

Model	Method	GSM8K	HumanEval	IFEval	AdvBench↓	LongBench
Qwen2.5-7B-Instruct	FP (bf16)	81.05	82.93	69.50	0.00	87.50
	Uniform	74.15±0.7	73.78±3.8	64.14±1.1	0.58±0.6	89.17±2.9
	PoT	53.95±4.4	58.94±2.5	63.65±1.9	0.83±1.0	84.17±2.9
	FPoT (Ours)	**71.65**±3.1	**70.33**±0.7	**66.91**±1.0	**0.32**±0.1	**84.17**±3.8
Llama-3.1-8B-Instruct	FP (bf16)	78.85	66.46	72.27	3.85	82.50
	Uniform	60.78±0.7	52.44±1.8	68.21±3.0	3.72±1.2	79.58±4.7
	PoT	44.58±1.6	27.64±1.5	57.48±2.2	9.55±3.4	61.77±16.5
	FPoT (Ours)	**57.67**±1.9	**47.76**±2.0	**66.73**±2.2	**6.03**±1.1	**79.44**±5.9

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim a 0.4%p 0-shot degradation versus the uniform INT baseline and an 8.72% relative gain over PoT on Llama-3-8B W4A4KV4, with results that hold across seven models and three bit-width configurations. Table 1 validates these numbers with mean$\pm$std over three calibration seeds.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 explicitly lists five limitations: (1) analytical HW cost model with open-source synthesis (Yosys) corroboration but without full RTL/ASIC validation, (2) per-weight-MAC efficiency figures whose end-to-end gain is largest in compute-bound prefill and smaller in memory-bound decode, (3) dependence on PTQ preprocessing, on which uniform INT at W4A4KV4 also relies, (4) single-GPU memory limits model scale to $\leq$14B, and (5) k=2 optimality is 4-bit specific.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix B.1 provides formal statements and proofs for Properties 1–4 of Normalized Bin Entropy (NBE): bounds, information-theoretic interpretation, affine invariance, and grid-structure sensitivity under fixed dynamic range; Property 5 (correlation with MSE for log-concave distributions) is supported by empirical justification, with the formal rate-distortion analysis stated as future work. The hardware cost model (Appendix E.1) is an analytical component-counting model with clearly stated assumptions (Remark 1 in Appendix E.1).

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 gives the preprocessing (OSTQuant, 400 steps with STE), the GPTQ weight quantization, the grid-alignment steps (G=16), the $\sqrt{2}$ approximation (1.5) and the evaluation protocol, and Appendix F gives the batch size and the fused Triton kernel. The GPTQ damping and column order, the optimizers and the calibration set are listed in the checklist item on experimental details below, and the learning rates are set in the code in the supplementary material.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is included in the supplementary material, with a README that gives installation steps and the commands that reproduce the main results. All experiments use publicly available models (Meta Llama-2/3, Mistral-7B, Qwen2.5) and datasets (WikiText-2, lm-eval-harness benchmarks).

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 5 and the code in the supplementary material specify the training and test details: OSTQuant training (400 steps, batch size 4, Riemannian Adam for rotation, standard Adam for scale), GPTQ (percdamp=1.0, act_order=False, per-channel), grid-alignment (G=16), calibration (WikiText-2 train, 128 samples, 2048 tokens), and evaluation (WikiText-2 PPL + 9-task zero-shot via lm-eval-harness [Gao et al., 2024]). Appendix G.4 and G.5 provide sensitivity analysis for zero-threshold τ and rounding mode.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports mean$_{\pm\text{std}}$ over three calibration seeds (0, 42, 123) for each of 126 cells (21 configurations [7 models $\times$ 3 bit-widths] $\times$ 3 weight quantization methods $\times$ 2 metrics). The preprocessing $\times$ quantizer factorial (Table 23) and the generation suite (Table 24) also report mean$\pm$std over the same three seeds. Other ablation and per-task tables report single-seed results to allocate the compute budget toward the main 21-configuration variance study.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on NVIDIA A100 80GB GPUs. OSTQuant training takes approximately 3–6 hours per model depending on size (7B–14B). GPTQ weight quantization takes approximately 10–30 minutes. The full experimental campaign (189 variance runs and ablations) required approximately 1,500 GPU-hours.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work focuses on improving the hardware efficiency of LLM quantization. It uses only publicly available pre-trained models and standard evaluation benchmarks, and does not involve human subjects, private data, or dual-use concerns beyond those inherent to LLMs in general.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 6 discusses the positive impact of enabling energy-efficient LLM inference through multiplier-free hardware, which can reduce the environmental footprint of LLM deployment. The negative societal impacts are those common to any LLM efficiency improvement—making LLMs cheaper to deploy could amplify both beneficial and harmful applications—but this work does not introduce new capabilities beyond cost reduction.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: This work releases a quantization method (code), not a new pre-trained model or dataset. The quantized models are functionally equivalent to the original publicly available models with slightly degraded accuracy.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models (Llama-2/3 under Meta Community License, Mistral-7B under Apache 2.0, Qwen2.5 under Apache 2.0) and datasets (WikiText-2 under CC-BY-SA 3.0) are cited with their original papers. The codebase builds on OSTQuant, which is cited and credited.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code in the supplementary material includes a README with installation instructions, example commands to reproduce main results, and documentation of all command-line arguments. The fused Triton kernel is documented in Appendix F.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology. The pre-trained LLMs used in experiments are standard evaluation targets, not part of the method itself.