TERNREFINE: GRADIENT-GUIDED FIXED-CAPACITY REFINEMENT OF TERNARY LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training ternarization produces a compact discrete model, but the resulting ternary assignment need not be optimal for the target task. We introduce TernRefine, a refinement method that reallocates side-state occupancy after quantization while preserving the ternary codebook and the exact number of side states in every quantization group. TernRefine computes a single task gradient at the deployed quantized model, uses the first-order score $-\langle G, \Delta Q \rangle$ to rank legal relocations, and selects how many relocations to apply using a separate validation split. We find that the point at which the gradient is evaluated matters substantially. In a controlled study over five fitting offsets, gradients computed at the quantized model and at the corresponding full-precision model produce patches with less than 2% Jaccard overlap and lead to opposite average outcomes on held-out data. Ranking with the quantized model gradient improves WikiText-2 in all five runs, whereas ranking with the full-precision gradient degrades it on average. TernRefine also improves WikiText-2 and C4 on OPT-350M, Llama-2-7B, and Qwen3-8B after affine ternarization. Starting from PT^2 models, all three fitted patches improve both datasets on Llama-2-7B and Qwen3-8B, and the selected models show positive average transfer on standard zero-shot tasks. Overall, these results show that ternary models can retain useful task structure after quantization and that gradients evaluated at the deployed discrete state provide an effective signal for refining that structure.

1 INTRODUCTION

Large language models (LLMs) require substantial memory and memory bandwidth at inference time, which has motivated increasingly aggressive weight quantization. Ternary representations restrict each weight to one of three codewords and have been studied through low-bit training, post-training conversion, calibration, and transformation-based quantization (Wang et al., 2023; Ma et al., 2024; Chen et al., 2024; Guo et al., 2025; Yan et al., 2026; Wang et al., 2026; Zhao et al., 2026). These methods have made ternary model construction increasingly effective. Yet constructing a quantizer and finding the best discrete assignment for a task are not the same problem. After quantization, the deployed ternary model may still admit discrete changes that reduce task loss.

We study this remaining freedom under a strict capacity constraint. Within each quantization group, a coordinate currently assigned to a side codeword can exchange roles with a coordinate assigned to the center codeword. The codebook and the number of side-state entries in each group remain unchanged. For affine ternarization, the center codeword need not be numerically zero, so the preserved quantity is the number of side states rather than ordinary sparsity. This gives a simple question: how much task performance can be recovered by rearranging an already quantized ternary model without changing its codebook or groupwise capacity?

TernRefine addresses this question in three steps. Capacity-Preserving Side-State Relocation (CPSR) defines the legal exchanges. Quantized-Point Gradient Projection (QGP) computes one task gradient at the deployed ternary model and ranks the legal changes using the first-order score $-\langle G, \Delta Q \rangle$. A separate validation split is then used to choose how far to follow this fixed ranking. The model is edited directly in its discrete state; TernRefine does not re-quantize the model or iteratively optimize latent full-precision weights.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A central question is where the task gradient should be measured. We therefore hold the quantized model, legal candidate set, task objective, and 384-relocation budget fixed, and change only the point at which the gradient is computed. Across five OPT-350M fitting offsets, ranking with the gradient of the quantized model improves WikiText-2 in all five runs, with mean ΔNLL -0.0627 ± 0.0337 on WikiText-2 and -0.0653 ± 0.0452 on C4. In contrast, using the corresponding gradient from the full-precision model degrades both datasets on average. The resulting patches also have less than 2% Jaccard overlap. Thus quantization changes which legal discrete directions appear useful to the task, and this difference is large enough to reverse the quality of a finite patch.

The same refinement principle also works beyond this controlled experiment. TernRefine improves held-out language-modeling loss on OPT-350M, Llama-2-7B, and Qwen3-8B after affine ternarization. Starting from PT^2 models, all three fitted patches improve both WikiText-2 and C4 on Llama-2-7B and Qwen3-8B. The selected models also improve the six-task standardized zero-shot macro score on both large models. These results indicate that ternary quantization can leave useful task information in the final discrete assignment that can still be recovered without changing the underlying ternary representation.

Our contributions are:

- **A fixed-capacity refinement problem after ternary PTQ.** We study discrete changes that preserve the ternary codebook and the exact number of side states in every quantization group, and show consistent held-out improvements across three model families and PT^2 initializations.
- **Evidence that the gradient evaluation point matters.** In a matched five-offset experiment, gradients computed at the quantized model and at the corresponding full-precision model produce almost disjoint patches and markedly different held-out results. This shows that the point at which task sensitivity is measured is important for finite discrete refinement.
- **A direct refinement procedure for the deployed ternary model.** CPSR defines the legal changes, QGP ranks them using one task gradient, and a separate validation split determines how many ranked changes to apply. The procedure uses one backward pass, requires no latent-weight optimization, and performs no re-ranking.

2 Related Work

Low-bit and ternary quantization for LLMs. Post-training quantization reduces deployment cost without full retraining. GPTQ/OPTQ uses approximate second-order information (Frantar et al., 2023), AWQ exploits activation-aware weight saliency (Lin et al., 2024), OmniQuant learns quantization parameters and equivalent transformations (Shao et al., 2024). QuIP uses incoherence processing (Chee et al., 2023), while QuIP# combines stronger incoherence processing with lattice codebooks (Tseng et al., 2024). Ternary representations push compression further. BitNet and BitNet b1.58 study extremely low-bit training (Wang et al., 2023; Ma et al., 2024), while TernaryLLM, PT-BitNet, PT^2-LLM, CAT-Q, and TWLA develop ternarization or post-training conversion procedures for pretrained models (Chen et al., 2024; Guo et al., 2025; Yan et al., 2026; Wang et al., 2026; Zhao et al., 2026). These methods primarily construct or calibrate the low-bit state. TernRefine instead starts after that construction stage and keeps the resulting ternary interface frozen.

Discrete refinement after quantization. ReQuant revisits discrete assignments after quantization while retaining the original grid, using reconstruction error to guide iterative refinement (Ma et al., 2026). TernRefine operates at the same post-quantization stage but studies a narrower fixed-cardinality ternary action space and ranks legal moves with task loss rather than reconstruction. Donor–receiver relocation is also conceptually related to fixed-sparsity prune-and-regrow methods such as RigL (Evci et al., 2020); however, the affine center codeword here need not be zero and the object being preserved is side-state cardinality, not numerical sparsity.

Task-aware quantization and quantized-point gradients. Task information has been used during rounding, quantizer construction, and quantization-aware optimization. AdaRound optimizes rounding decisions (Nagel et al., 2020), BRECQ uses block reconstruction (Li et al., 2021), SqueezeLLM incorporates sensitivity into low-bit quantization (Kim et al., 2024), and SignRound uses signed

gradient descent to optimize rounding and clipping (Cheng et al., 2024). GuidedQuant and GTAQ instead inject task-level gradient information during quantizer construction or PTQ (Kim et al., 2025; Shao et al., 2026). Quantization-aware training such as LLM-QAT adapts the model under simulated low-bit quantization (Liu et al., 2024). Most closely related to our evaluation-point question, Li et al. (2026) analyze why gradients evaluated at quantized weights can contain useful low-loss information in QAT. TernRefine brings this sensitivity signal to a different stage and action space: the ternary state is already constructed and frozen, no latent or continuous quantizer variable is optimized, and one task gradient is projected directly onto a fixed-grid, exact-cardinality discrete neighborhood. Our contribution is the resulting post-PTQ relocation problem and the matched evidence that the gradient evaluation point changes finite-patch quality inside that neighborhood.

3 PRELIMINARIES

3.1 TERNARY POST-TRAINING QUANTIZATION

Let Q denote a ternary model produced by a post-training quantization procedure. For each quantization group g, we consider the affine ternary codebook

$$\mathcal{C}_g = \{\mu_g - \alpha_g,\ \mu_g,\ \mu_g + \alpha_g\}, \qquad \alpha_g > 0, \tag{1}$$

with the equivalent index representation

$$Q_i = \mu_g + \alpha_g T_i, \qquad T_i \in \{-1, 0, +1\}. \tag{2}$$

Here, Q_i denotes the quantized value of coordinate i in group g, and T_i specifies which of the three codewords is assigned to that coordinate. Centered ternarization is the special case $\mu_g = 0$. In the affine case, $T_i = 0$ denotes the center codeword μ_g, which need not be numerically zero.

We refer to coordinates with $T_i \neq 0$ as occupying a *side state*. TernRefine keeps the codebook $\mathcal{C}_g$ fixed and preserves the exact number of side-state coordinates in every quantization group throughout refinement.

3.2 TASK SENSITIVITY AT THE QUANTIZED MODEL

Let $\mathcal{L}(Q)$ denote a task loss evaluated on the quantized model. The gradient

$$\nabla_Q \mathcal{L}(Q) \tag{3}$$

measures the local sensitivity of the loss to changes in the deployed quantized weights. Recent work has shown that gradients evaluated at quantized weights can contain useful information about nearby low-loss directions (Li et al., 2026).

TernRefine uses this gradient after the ternary model has already been constructed and frozen. It does not optimize a continuous quantizer or latent full-precision weights. Instead, the gradient is used to rank feasible discrete changes to the existing ternary assignment. The precise relocation rule and its first-order score are introduced in Sec. 4.3.

4 METHOD

Given a frozen ternary state Q_0, TernRefine restricts refinement to legal side–center exchanges while keeping the codebook, group structure, side labels, and initializer-specific transforms or permutations fixed. One task gradient at Q_0 ranks these exchanges, and a disjoint validation split selects how many ranked changes to apply. We call the three stages Capacity-Preserving Side-State Relocation (CPSR), Quantized-Point Gradient Projection (QGP), and Adaptive Patch Growth (APG). Figure 1 and Algorithm 1 summarize the procedure.

4.1 FIXED-CAPACITY DISCRETE REFINEMENT

Let Q_0 denote a frozen deployed ternary model represented as in Eq. (2). A compatible initializer exposes the group assignment and ternary index T_i, the frozen codebook (μ_g, α_g), the candidate side-state label of each center-coded receiver, and any frozen transforms or permutations needed to

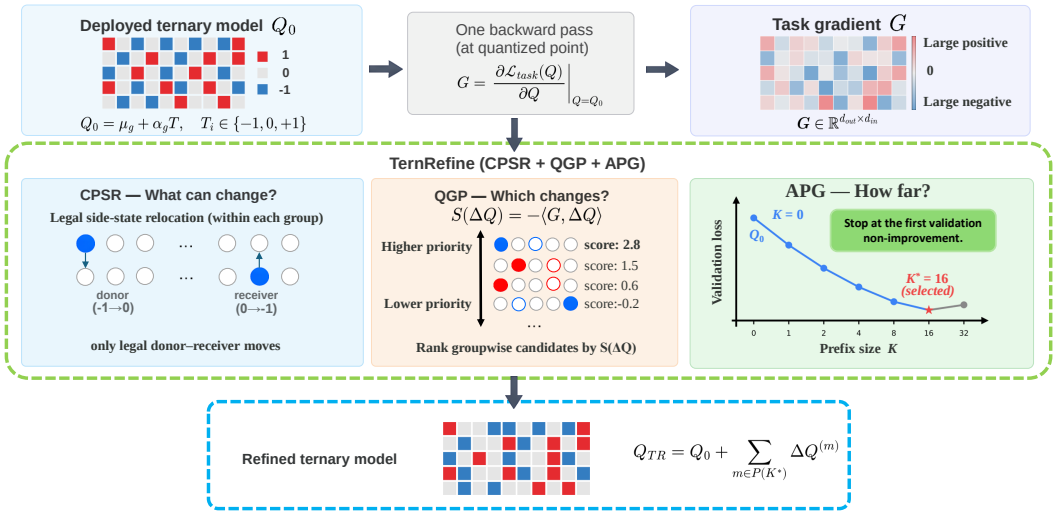


Figure 1: **Overview of TernRefine.** Starting from a frozen ternary state Q_0, one task gradient ranks legal groupwise relocations, and a disjoint selection split determines how far to follow this fixed ranking. The refined state Q_{TR} preserves the ternary codebook and the exact side-state cardinality of every quantization group.

interpret deployed coordinates. These quantities remain unchanged throughout refinement; TernRefine modifies only the allocation of side-state occupancy. Throughout the paper, *fixed capacity* means that the number of side-state entries in every quantization group is preserved.

Let

$$\mathcal{R}_{\mathrm{legal}}(Q_0) = \{m_1, \ldots, m_{N_{\mathrm{move}}}\} \tag{4}$$

denote the legal relocations defined by CPSR, where each move m induces a sparse perturbation $\Delta Q^{(m)}$. For binary selections $a_m \in \{0, 1\}$, the resulting state is

$$Q(\mathbf{a}) = Q_0 + \sum_{m \in \mathcal{R}_{\mathrm{legal}}(Q_0)} a_m \Delta Q^{(m)}. \tag{5}$$

Let $\Gamma_{im} = 1$ when move m modifies coordinate i, and $\Gamma_{im} = 0$ otherwise. Given a relocation budget $B_{\max}$, the corresponding discrete optimization problem is

$$\begin{aligned} \mathbf{a}^\star \in \arg\min_{\mathbf{a}} \quad & \mathcal{L}_{\mathrm{fit}}\left(Q_0 + \sum_m a_m \Delta Q^{(m)}\right) \\ \text{s.t.} \quad & a_m \in \{0, 1\}, \qquad \sum_m a_m \le B_{\max}, \\ & \sum_m \Gamma_{im} a_m \le 1, \qquad \forall i. \end{aligned} \tag{6}$$

Here

$$\mathcal{L}_{\mathrm{fit}}(Q) = \frac{1}{|\mathcal{D}_{\mathrm{fit}}|} \sum_{(x,y) \in \mathcal{D}_{\mathrm{fit}}} \ell_{\mathrm{CE}}(f_Q(x), y)\,. \tag{7}$$

Each CPSR relocation changes exactly two ternary coordinates, so $B_{\max}$ counts relocations rather than changed coordinates. Exact solution of Eq. (6) is combinatorial. TernRefine instead uses a structured approximation: CPSR defines the feasible edits, QGP retains one locally preferred relocation from each quantization group and ranks these candidates, and APG selects a layer-balanced prefix of the resulting fixed ranking.

4.2 Capacity-Preserving Side-State Relocation (CPSR)

Using the ternary index representation in Eq. (2), write

$$T_i = \sigma_i z_i, \qquad \sigma_i \in \{-1, +1\}, \qquad z_i \in \{0, 1\}. \tag{8}$$

Here $z_i = 0$ denotes the center state and $z_i = 1$ a side state. For an occupied side-state coordinate, σ_i is its current ternary sign. For a center-state coordinate, σ_i is the candidate side-state label supplied by the frozen initializer metadata. Thus TernRefine changes the occupancy variable z_i while leaving the side-label rule and codebook fixed.

For group g with coordinate set I_g, define

$$A_g = \{i \in I_g : z_i = 1\}, \qquad C_g = \{i \in I_g : z_i = 0\}, \tag{9}$$

with groupwise side-state cardinality

$$k_g = \sum_{i \in I_g} z_i. \tag{10}$$

A CPSR move $m = (d, r)$ selects a donor $d \in A_g$ and a receiver $r \in C_g$ within the same group and applies

$$z_d : 1 \to 0, \qquad z_r : 0 \to 1. \tag{11}$$

The move preserves k_g exactly. Since $Q_i = \mu_g + \alpha_g T_i$, its numerical perturbation is

$$\Delta Q_d = -\alpha_g \sigma_d, \qquad \Delta Q_r = +\alpha_g \sigma_r. \tag{12}$$

Neither μ_g nor α_g changes. Each relocation modifies only its donor and receiver coordinates. A patch is compatible when no coordinate participates in more than one relocation; such a patch preserves both the codebook and the side-state cardinality of every group.

4.3 Quantized-Point Gradient Projection (QGP)

CPSR defines the feasible directions but does not identify which relocations are useful for the task. QGP measures task sensitivity at the deployed ternary state using one gradient on the fitting split,

$$G = \nabla_Q \mathcal{L}_{\text{fit}}(Q)\big|_{Q=Q_0}. \tag{13}$$

For a composite perturbation $\Delta Q(\mathbf{a}) = \sum_m a_m \Delta Q^{(m)}$, a first-order expansion around Q_0 gives

$$\mathcal{L}_{\text{fit}}\left(Q_0 + \Delta Q(\mathbf{a})\right) = \mathcal{L}_{\text{fit}}(Q_0) + \langle G, \Delta Q(\mathbf{a})\rangle + \mathcal{O}\big(\|\Delta Q(\mathbf{a})\|_F^2\big) \tag{14}$$

$$\approx \mathcal{L}_{\text{fit}}(Q_0) - \sum_m a_m S_m, \tag{15}$$

where

$$S_m = -\left\langle G, \Delta Q^{(m)} \right\rangle. \tag{16}$$

Thus $S_m > 0$ predicts a first-order loss decrease at Q_0. Because CPSR edits are applied to Q_0, this is the first-order sensitivity at the state being edited; finite patches can still introduce higher-order interactions.

For a donor–receiver relocation $m = (d, r)$, substituting Eq. (12) yields

$$S(d, r) = -\left(G_d \Delta Q_d + G_r \Delta Q_r\right) \tag{17}$$

$$= \alpha_g \left(G_d \sigma_d - G_r \sigma_r\right). \tag{18}$$

The affine offset μ_g cancels from the perturbation, so the same expression applies to centered and affine ternary codebooks.

Pairing invariance and groupwise projection. For donor set $D \subseteq A_g$, receiver set $R \subseteq C_g$ with $|D| = |R|$, and any bijection $\pi : D \to R$,

$$\sum_{d \in D} S\big(d, \pi(d)\big) = \alpha_g \left(\sum_{d \in D} G_d \sigma_d - \sum_{r \in R} G_r \sigma_r\right). \tag{19}$$

Thus the first-order objective separates into donor and receiver terms. For a fixed number of relocations, it favors donors with large $G_d \sigma_d$ and receivers with small $G_r \sigma_r$; the bijective pairing itself does not change the summed surrogate score.

For the model-scale procedure, we retain one such relocation per eligible group. The best single relocation is obtained by

$$\begin{aligned} d_g^\star &= \arg\max_{d \in A_g} \alpha_g G_d \sigma_d, \\ r_g^\star &= \arg\max_{r \in C_g} \left(-\alpha_g G_r \sigma_r\right), \\ m_g^\star &= (d_g^\star, r_g^\star). \end{aligned} \tag{20}$$

This construction reduces candidate formation from pairwise donor–receiver enumeration to two groupwise extremum operations. Within each eligible decoder layer $\ell \in \mathcal{M}$, the projected moves are sorted once by decreasing QGP score to obtain the fixed ranking

$$\mathcal{R}_\ell = (m_{\ell,1}, m_{\ell,2}, \ldots), \qquad S_{m_{\ell,1}} \geq S_{m_{\ell,2}} \geq \cdots . \tag{21}$$

Because at most one relocation is retained from each disjoint quantization group, this model-scale ranking is coordinate-compatible by construction. QGP is computed once at Q_0 and is never updated during patch growth.

4.4 Adaptive Patch Growth (APG)

QGP gives a local ordering at Q_0, but its accuracy can degrade as relocations accumulate. We therefore use a disjoint selection split to choose the prefix length.

For a tested per-layer prefix size K that is feasible in every eligible layer, define the layer-balanced patch

$$\mathcal{P}(K) = \bigcup_{\ell \in \mathcal{M}} \{m_{\ell,1}, \ldots, m_{\ell,K}\}, \tag{22}$$

and reconstruct its state from the same frozen deployment,

$$Q^{(K)} = Q_0 + \sum_{m \in \mathcal{P}(K)} \Delta Q^{(m)}. \tag{23}$$

Since every relocation changes two coordinates, a prefix of size K per layer contains $K|\mathcal{M}|$ relocations and changes $2K|\mathcal{M}|$ coordinates.

Let

$$0 = K_0 < K_1 < \cdots < K_J, \qquad Q^{(0)} = Q_0,$$

be the tested prefix schedule. Patch extent is selected using

$$\mathcal{L}_{\text{sel}}(Q) = \frac{1}{|\mathcal{D}_{\text{sel}}|} \sum_{(x,y) \in \mathcal{D}_{\text{sel}}} \ell_{\text{CE}}(f_Q(x), y), \tag{24}$$

where $\mathcal{D}_{\text{fit}} \cap \mathcal{D}_{\text{sel}} = \varnothing$. Define the first non-improvement index

$$j^\star = \min\left\{ j \in \{0, \ldots, J-1\} : \mathcal{L}_{\text{sel}}\left(Q^{(K_{j+1})}\right) \geq \mathcal{L}_{\text{sel}}\left(Q^{(K_j)}\right) \right\}. \tag{25}$$

When such an index exists, APG returns the preceding prefix,

$$K^\star = K_{j^\star}, \qquad Q_{\text{TR}} = Q^{(K^\star)}. \tag{26}$$

If all tested prefixes improve, APG returns $K^\star = K_J$.

QGP requires one fitting backward pass. APG then evaluates several prefixes by forward passes only; each prefix is reconstructed from Q_0, and the scores are never recomputed.

5 Experiments

5.1 Experimental Setup

Models and evaluation. We evaluate OPT-350M, Llama-2-7B, and Qwen3-8B (Zhang et al., 2022; Touvron et al., 2023; Yang et al., 2025). Held-out language modeling quality uses WikiText-2 (W2) and C4 (Merity et al., 2016; Raffel et al., 2020); both are disjoint from all fitting and selection

Algorithm 1 TERNREFINE: fixed-capacity ternary refinement.

Require: Frozen state Q_0; eligible layers $\mathcal{M}$; fitting split $\mathcal{D}_{\mathrm{fit}}$; selection split $\mathcal{D}_{\mathrm{sel}}$; schedule $0 = K_0 < K_1 < \cdots < K_J$
Ensure: Refined ternary state Q_{TR}

$G \leftarrow \nabla_Q \mathcal{L}_{\mathrm{fit}}(Q)|_{Q=Q_0}$
for each layer $\ell \in \mathcal{M}$ **do**
    Project each eligible group g to its best CPSR move $m_g^\star$ using Eq. (20)
    Sort $\{m_g^\star\}$ by QGP score to obtain fixed ranking $\mathcal{R}_\ell$
end for
$L_{\mathrm{prev}} \leftarrow \mathcal{L}_{\mathrm{sel}}(Q_0)$; $Q_{\mathrm{TR}} \leftarrow Q_0$
for $j = 1, \ldots, J$ **do**
    $\mathcal{P}(K_j) \leftarrow \bigcup_{\ell \in \mathcal{M}} \mathcal{R}_\ell[1{:}K_j]$
    $Q^{(K_j)} \leftarrow \mathrm{Apply}(Q_0, \mathcal{P}(K_j))$
    $L_j \leftarrow \mathcal{L}_{\mathrm{sel}}(Q^{(K_j)})$
    if $L_j \geq L_{\mathrm{prev}}$ **then**
        break
    end if
    $Q_{\mathrm{TR}} \leftarrow Q^{(K_j)}$; $L_{\mathrm{prev}} \leftarrow L_j$
end for
return Q_{TR}

data. We report NLL unless noted (lower is better), and use PPL for the PT^2 summary. Zero-shot transfer uses `lm-evaluation-harness` (Gao et al., 2024) on HellaSwag, PIQA, ARC-Easy, ARC-Challenge, WinoGrande, and MMLU (Zellers et al., 2019; Bisk et al., 2020; Clark et al., 2018; Sakaguchi et al., 2020; Hendrycks et al., 2021). All held-out language modeling and downstream evaluations are excluded from fitting and APG selection. Full-precision FP16/BF16 endpoints use the same W2/C4 evaluator paths, and the Llama/Qwen FP references use the same downstream metric convention. These rows provide a common degradation reference; refinement methods are compared within the same quantized initializer and protocol.

Canonical refinement protocol and naming. The ordinary affine experiments use group size 128 with frozen (μ_g, α_g) and refine attention Q/K projections. Candidate scoring uses one CE backward pass on $\mathcal{D}_{\mathrm{fit}}$; APG evaluates the fixed ranking on a disjoint $\mathcal{D}_{\mathrm{sel}}$ with $K \in \{1, 2, 4, 8, 16, 32, 64\}$ unless the budget is fixed. We call these settings *TernRefine-Fixed* and *TernRefine-APG*, respectively. For the OPT mechanism study, a fitting offset denotes the token starting index of a fitting window in the tokenized WikiText-2 training stream. Exact offsets and PT^2 fit/selection indices are given in Appendix A.1.

Matched fixed-budget controls and validation-selected runs are compared only within protocol families: fixed-budget studies isolate mechanisms under matched edit counts, whereas TernRefine-APG reports validation-selected endpoints. The full protocol map is in Appendix A.1.

5.2 MAIN EFFECTIVENESS ACROSS MODELS AND PTQ INITIALIZERS

Table 1 evaluates constrained refinement across model families and PTQ initializer strengths.

TernRefine-Fixed lowers W2/C4 on all three ordinary affine models while matched random remains near baseline. After PT^2, all three fits improve both datasets for Llama-2-7B and Qwen3-8B; the corresponding NLL gap recovery is 10.25%/9.32% and 24.93%/20.27% on W2/C4, respectively (Appendix A.8). Fig. 2 schematically illustrates the lower beneficial-move density after PT^2.

The recovery rows report the fraction of the matched FP-to-PT^2 NLL gap recovered by TernRefine. ReQuant and module scope controls are reported in Appendices A.6 and A.4.

5.3 GRADIENT EVALUATION POINT AND GUIDANCE EVIDENCE

Table 2 compares Q-point and FP-point rankings on five matched OPT-350M offsets and three prespecified Llama-2-7B fits, holding the deployed state, legal proposal pool, fitting loss, and per-layer edit density fixed. For OPT, the Reconstruction control re-ranks the same legal pool by the

Table 1: Language modeling results across model families and PTQ initializers. Lower is better; bold marks the best matched quantized result. PT^2 + TernRefine reports mean ± sample std over fits 0/17/33; FP rows are full-precision references.

Init.	Metric	Dataset	Method	OPT-350M	Llama-2-7B	Qwen3-8B
FP ref.	NLL	W2	FP16/BF16	3.0913	1.6998	2.2747
	NLL	C4	FP16/BF16	3.1175	1.9833	2.7363
Affine	NLL	W2	baseline	4.3813	7.6208	4.5841
			matched random	4.3810	7.6215	4.5846
			TernRefine-Fixed	**4.3099**	**7.5501**	**4.5169**
	NLL	C4	baseline	3.7676	7.5668	4.6619
			matched random	3.7682	7.5633	4.6583
			TernRefine-Fixed	**3.6736**	**7.4018**	**4.5655**
PT^2	PPL	W2	baseline	–	11.6430	49.204
			+ TernRefine-APG	–	**10.7769±0.1386**	**32.844±0.426**
	PPL	C4	baseline	–	24.3244	175.976
			+ TernRefine-APG	–	**21.7451±0.8101**	**107.441±1.916**
	Recovery	W2	NLL gap recovered	–	10.25%	24.93%
		C4	NLL gap recovered	–	9.32%	20.27%

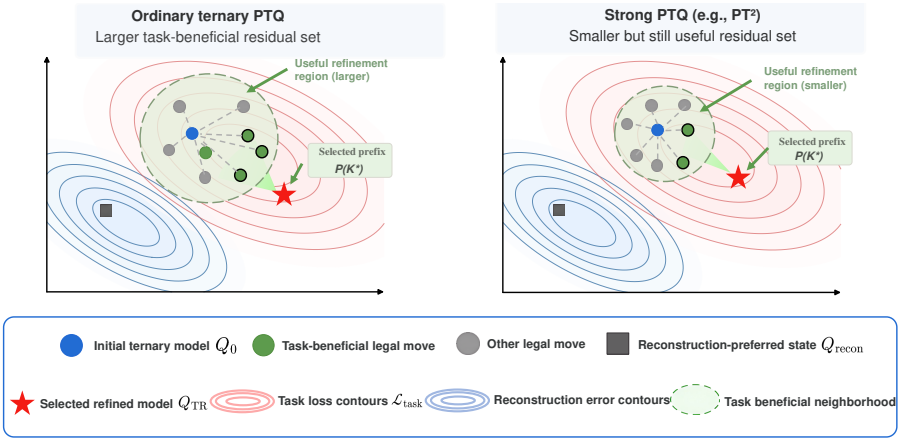


Figure 2: Schematic residual refinement after ternary PTQ. On Llama-2-7B, beneficial-move density among ranked W2 relocations drops from 68.36% (affine) to 32.42% (PT^2), while useful moves remain (Appendix A.8).

exact donor/receiver full-precision weight squared-error decrease; Random uses the same relocation budget. Full definitions and per-offset controls are reported in Appendix A.2.

Table 2: Matched evaluation-point ablations on OPT-350M and Llama-2-7B. Values are mean ± sample std ΔNLL; "Wins" counts improvements on Val/W2/C4.

Model	Ranking	ΔVal	ΔW2	ΔC4	Wins
OPT-350M	**QGP (quant. point)**	**−0.0631±0.0625**	**−0.0627±0.0337**	**−0.0653±0.0452**	**4/5/4**
	FP-point task grad.	+0.1201±0.0630	+0.1075±0.0669	+0.1082±0.0663	0/0/1
	Reconstruction	−0.0025±0.0021	−0.0010±0.0008	−0.0020±0.0016	5/5/5
	Random	−0.0034±0.0025	−0.0030±0.0018	−0.0027±0.0019	5/5/5
Llama-2-7B	**QGP (quant. point)**	**−0.0560±0.0215**	**−0.0682±0.0505**	−0.0043±0.0625	**3/3/1**
	FP-point task grad.	+0.0013±0.0058	+0.0060±0.0108	+0.0110±0.0112	2/1/1

On OPT, QGP improves W2 on all five offsets, whereas FP-point degrades it on average; the selected patches have Jaccard overlap 0.0039–0.0199. On Llama-2-7B, Q-point improves Val/W2 in all three fits while C4 is mixed. The large-model comparison uses the same per-layer density (16

relocations/layer; 512 total) rather than the OPT absolute budget. Full per-split, candidate, and allocation diagnostics are reported in Appendices A.2–A.3.

5.4 Adaptive Composition

APG uses validation loss to select the useful range as a fixed first-order ranking accumulates relocations (Table 3).

Table 3: Adaptive patch growth on OPT-350M: mean ± sample std ΔNLL; lower is better.

Budget rule	ΔVal	ΔW2	ΔC4
Fixed $K = 16$	-0.3060 ± 0.0151	-0.4134 ± 0.0111	-0.1660 ± 0.0165
Fixed $K = 32$	-0.3025 ± 0.0378	$\mathbf{-0.4355 \pm 0.0170}$	-0.1411 ± 0.0381
Fixed $K = 64$	-0.2389 ± 0.0782	-0.4230 ± 0.0424	-0.0461 ± 0.0763
APG	$\mathbf{-0.3150 \pm 0.0281}$	-0.4252 ± 0.0170	$\mathbf{-0.1671 \pm 0.0176}$

APG selects $K^\star = 16, 32, 16$, giving the best mean validation and C4; fixed $K = 32$ is slightly better on held-out W2, which is not used for selection. On Qwen3-8B, validation forwards account for 93% of the 116-s pipeline time (Appendix A.5).

5.5 Standardized Downstream Transfer

We evaluate the preselected PT^2 endpoints on six standardized zero-shot tasks. No downstream task is used for fitting, APG selection, or endpoint reselection.

Table 4: Zero-shot transfer after PT^2. +TR reports mean ± std over three fitted endpoints; FP is the full-precision reference. Chance references, per-fit results, and paired statistics are in Appendix A.7.

Task	Llama FP	Llama PT^2	Llama + TR	Qwen FP	Qwen PT^2	Qwen + TR
HellaSwag	75.99	41.54	$\mathbf{44.81 \pm 0.58}$	74.94	30.32	$\mathbf{32.19 \pm 0.42}$
PIQA	79.27	62.68	$\mathbf{64.40 \pm 0.77}$	77.69	56.20	$\mathbf{56.57 \pm 0.60}$
ARC-Easy	74.62	42.09	$\mathbf{44.36 \pm 0.04}$	80.68	33.46	$\mathbf{37.46 \pm 0.46}$
ARC-Challenge	46.33	23.89	$\mathbf{25.82 \pm 0.21}$	56.74	23.38	$\mathbf{24.91 \pm 0.48}$
WinoGrande	69.06	**58.72**	57.09 ± 0.78	68.11	51.78	$\mathbf{52.33 \pm 1.11}$
MMLU	40.78	23.02	$\mathbf{23.30 \pm 0.20}$	72.95	22.90	$\mathbf{22.95 \pm 0.05}$
Macro	64.34	41.99	$\mathbf{43.30 \pm 0.36}$	71.85	36.34	$\mathbf{37.74 \pm 0.24}$

Macro accuracy increases on both models, with task-dependent gains and a substantial FP gap. Every Llama endpoint improves five of six tasks, with WinoGrande the only consistent decrease; the Qwen endpoint win/loss counts are 5/1, 6/0, and 5/1. For Llama-2-7B, the paired macro shift is $+1.309$ pp (95% CI $[+0.895, +1.719]$); details are in Appendix A.7.

6 Conclusion

TernRefine treats post-PTQ ternary assignment as a fixed-capacity discrete refinement problem and uses deployed-state gradients to rank legal relocations. Across three model families, this improves held-out language-modeling loss, including after PT^2 on Llama-2-7B and Qwen3-8B. Matched Q-point/FP-point controls further show that the gradient evaluation point materially changes finite-patch quality, revealing useful task structure that remains in the discrete assignment after ternary PTQ.

AI use statement

Generative AI tools were used to assist with experimental design, including brainstorming and refining controlled comparisons, ablation structure, evaluation settings, and validation procedures. They were also used to assist with manuscript writing and polishing, literature-search support, experimental result checking and analysis, code/debugging support, and figure/formatting work. All AI-assisted suggestions were reviewed by the authors; the authors made the final methodological and experimental decisions and independently checked the reported technical content, citations, and

results. The authors take responsibility for the final content of this work, including text, claims, experimental design, and other artifacts produced with the aid of generative AI.

Reproducibility statement

Section 4 specifies the frozen-state interface, legal relocations, QGP scoring, pairing invariance, and the APG stopping rule. Section 5 separates controlled fixed-budget studies from the complete TernRefine pipeline. The appendix reports per-offset results, candidate diagnostics, module-scope experiments, APG trajectories, runtime, downstream evaluations, and state and legality checks. The anonymous supplementary artifact provides the exact candidate-construction procedure, deterministic conflict handling, and tie-breaking rules used in the reported experiments.

References

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, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M. De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/0df38cd13520747e1e64e5b123a78ef8-Abstract-Conference.html.

Tianqi Chen, Zhe Li, Weixiang Xu, Zeyu Zhu, Dong Li, Lu Tian, Emad Barsoum, Peisong Wang, and Jian Cheng. TernaryLLM: Ternarized large language model. *arXiv preprint arXiv:2406.07177*, 2024. doi: 10.48550/arXiv.2406.07177.

Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Kaokao Lv, and Yi Liu. Optimize weight rounding via signed gradient descent for the quantization of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11332–11350, 2024. doi: 10.18653/v1/2024.findings-emnlp.662.

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. doi: 10.48550/arXiv.1803.05457.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2943–2952. PMLR, 2020. URL https://proceedings.mlr.press/v119/evci20a.html.

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

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. The language model evaluation harness, 2024. URL https://zenodo.org/records/12608602.

Yufei Guo, Zecheng Hao, Jiahang Shao, Jie Zhou, Xiaode Liu, Xin Tong, Yuhan Zhang, Yuanpei Chen, Weihang Peng, and Zhe Ma. PT-BitNet: Scaling up the 1-bit large language model with post-training quantization. *Neural Networks*, 2025. doi: 10.1016/j.neunet.2025.107855.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens J. S. Schaefer, Deokjae Lee, Yeonhong Park, Jae W. Lee, and Hyun Oh Song. GuidedQuant: Large language model quantization via exploiting end loss guidance. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30011–30037. PMLR, 2025. URL https://proceedings.mlr.press/v267/kim25d.html.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23901–23923. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24f.html.

Hanyang Li, Jianhao Ma, and Ying Cui. Understanding quantization-aware training: Gradients at quantized weights bias to the low-loss basin. *arXiv preprint arXiv:2606.09012*, 2026. doi: 10.48550/arXiv.2606.09012.

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

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.26. URL https://aclanthology.org/2024.findings-acl.26/.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit LLMs: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. doi: 10.48550/arXiv.2402.17764.

Yongge Ma, Guoan Wang, Feiyu Wang, Yaoming Li, Qian Zhang, Zihan Yan, Yinjun Han, and Tong Yang. ReQuant: Fixed-grid discrete refinement for post-training quantization. *arXiv preprint arXiv:2608.07019*, 2026. doi: 10.48550/arXiv.2608.07019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. doi: 10.48550/arXiv.1609.07843.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7197–7206. PMLR, 2020. URL https://proceedings.mlr.press/v119/nagel20a.html.

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. URL https://jmlr.org/papers/v21/20-074.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

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

Yihua Shao, Yan Gu, Minxi Yan, Siyu Chen, Haiyang Liu, Ziyang Yan, Yongjia Li, Yan Wang, Qun Song, Hao Tang, Haotong Qin, Jingcai Guo, and Nicu Sebe. Gradient enhancement task aware post-training quantization. In *Proceedings of the 35th International Joint Conference on Artificial Intelligence*, 2026.

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. doi: 10.48550/arXiv.2307.09288.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48630–48656. PMLR, 2024. URL https://proceedings.mlr.press/v235/tseng24a.html.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023. doi: 10.48550/arXiv.2310.11453.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, and Anbang Yao. CAT-Q: Cost-efficient and accurate ternary quantization for LLMs. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.26650.

Xianglong Yan, Chengzhu Bao, Zhiteng Li, Tianao Zhang, Kaicheng Yang, Haotong Qin, Ruobing Xie, Xingwu Sun, and Yulun Zhang. PT^2-LLM: Post-training ternarization for large language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.03267.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388.

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*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. doi: 10.48550/arXiv.2205.01068.

Zhixiong Zhao, Zukang Xu, Zhixuan Chen, Xing Hu, Zhe Jiang, and Dawei Yang. TWLA: Achieving ternary weights and low-bit activations for LLMs via post-training quantization. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.13054.

A Additional Experimental Results

A.1 Implementation and Protocol Notes

We distinguish controlled fixed-budget studies from the full validation-selected pipeline throughout the paper. *TernRefine-Fixed* uses a predetermined number of relocations for matched mechanism or transfer controls; *TernRefine-APG* denotes the complete pipeline with validation-selected prefix length. In the canonical model-scale path, each eligible quantization group is projected to its best single first-order relocation using Eq. (20), and these groupwise candidates are sorted once within each layer. Because groups are disjoint and at most one relocation is retained per group, the resulting ranking is coordinate-compatible by construction. In the matched OPT evaluation-point control, quantized-point and FP-point groupwise proposals are merged into one shared legal proposal pool, every objective re-scores the same pool, and coordinate conflicts are filtered deterministically when

needed. The anonymous supplementary artifact provides the exact proposal construction and tie-breaking conventions. The ranking remains fixed after construction.

For the matched OPT evaluation-point study, fitting offsets $\{0, 1024, 2048, 3072, 4096\}$ are token starting indices in the concatenated tokenized WikiText-2 training stream; every ranking uses the same 384-relocation budget (16/layer). The large-model replication uses Llama-2-7B fits 0/17/33 at nominal fitting-token offsets 0/34816/67584 and the same layer density, $K = 16$/layer, giving 512 relocations per endpoint. Q-point and FP-point arms share the same affine Q_0, fitting data, held-out sets, and legal CPSR proposal pool within each split. For the PT^2 study, fits 0/17/33 use the same nominal offsets and a fixed disjoint selection split with indices 1–16. Standard downstream runs use zero-shot `lm-evaluation-harness`, batch size 1, and maximum sequence length 2048. Full-precision FP16/BF16 references use the same W2/C4 evaluator path and downstream metric convention. Fitting, selection, held-out language modeling, and downstream data follow the disjoint roles defined in the main text.

For ordinary affine initialization, the candidate side label for a center-coded coordinate follows the pre-quantization residual sign: $\sigma_i = +1$ when $W_i^{\mathrm{FP}} - \mu_g \geq 0$ and -1 otherwise. For PT^2, the side label and deployed-coordinate interpretation come from the exported ternary sidecar and frozen SSR/permutation metadata.

Table 5: Experimental protocol map. Absolute values are compared only within a matched protocol family.

Study	Start / scope	Patch rule	Question
Affine	affine ternary, Q/K	TR-Fixed: 384 reloc.	cross-model LM gain
Eval. point	OPT/Llama, shared CPSR pool	TR-Fixed: 16/layer	quantized vs. FP gradient
Module scope	affine OPT, multi-module	TR-Fixed: 384 reloc.	module dependence
Composition	direct-ternary OPT	APG / fixed K	finite-patch extent
Strong PT^2	Llama/Qwen, Q/K	TR-APG	after optimized PTQ
FP refs.	FP16/BF16 checkpoints	no quantization	degradation reference
Downstream	frozen PT^2 endpoints	preselected endpoints	zero-shot transfer

A.2 Matched Quantized-Point vs. FP-Point Gradient Ablation

This matched ablation isolates the task-gradient evaluation point while holding the deployed ternary state, fitting objective, shared legal CPSR proposal pool, and edit budget fixed. We use OPT-350M fitting offsets $\{0, 1024, 2048, 3072, 4096\}$. Each ranking selects 384 legal relocations (768 changed coordinates) with zero group-cardinality violations. The Reconstruction ranking measures the exact decrease in FP-weight squared error on the two coordinates changed by each legal relocation,

$$S_{\mathrm{rec}}(m) = \sum_{i \in \{d,r\}} (Q_{0,i} - W_i^{\mathrm{FP}})^2 - \sum_{i \in \{d,r\}} (Q_{0,i} + \Delta Q_i^{(m)} - W_i^{\mathrm{FP}})^2, \tag{27}$$

without changing the shared action space or edit budget.

Table 6: Full five-offset matched evaluation-point results on OPT-350M. Values are ΔNLL relative to the shared affine Q_0; negative is better.

Offset	Ranking	ΔVal	ΔW2	ΔC4
0	QGP (quantized point)	-0.0690	-0.0605	-0.0868
	FP-point task grad.	+0.0162	+0.0013	-0.0054
	Reconstruction	-0.0048	-0.0004	-0.0030
	Random	-0.0075	-0.0051	-0.0058
1024	QGP (quantized point)	+0.0357	-0.0093	+0.0142
	FP-point task grad.	+0.1655	+0.1717	+0.1675
	Reconstruction	-0.0007	-0.0007	-0.0007
	Random	-0.0008	-0.0006	-0.0009
2048	QGP (quantized point)	-0.0746	-0.0708	-0.0715
	FP-point task grad.	+0.1373	+0.1452	+0.1196
	Reconstruction	-0.0008	-0.0006	-0.0000
	Random	-0.0030	-0.0032	-0.0018
3072	QGP (quantized point)	-0.0692	-0.0706	-0.0892
	FP-point task grad.	+0.1099	+0.0863	+0.1232
	Reconstruction	-0.0015	-0.0010	-0.0036
	Random	-0.0034	-0.0042	-0.0027
4096	QGP (quantized point)	-0.1383	-0.1022	-0.0934
	FP-point task grad.	+0.1714	+0.1330	+0.1362
	Reconstruction	-0.0046	-0.0024	-0.0029
	Random	-0.0025	-0.0019	-0.0022

The selected QGP and FP-point patches have Jaccard overlap 0.0039–0.0199 across the five offsets, while QGP and reconstruction have overlap 0–0.0013. Thus the matched change in evaluation point produces materially different legal patches.

Table 7: Candidate-level rank diagnostics from 512 shared legal moves per offset. Correlations are measured against realized single-move gains.

Statistic	QGP	FP-point	Reconstruction
ρ vs. realized fit gain	0.0920 ± 0.0593	0.0141 ± 0.0596	0.0244 ± 0.0399
ρ vs. realized Val gain	0.0460 ± 0.0638	0.0810 ± 0.0682	–

Table 8: Fit-domain beneficial precision among top-ranked sampled moves. A beneficial move lowers realized NLL.

Top fraction	QGP	FP-point	Reconstruction
1%	0.600	0.440	0.520
5%	0.569	0.477	0.500
10%	0.541	0.471	0.533
20%	0.543	0.522	0.527

Across the sampled candidates, the QGP and FP-point score rankings have mean Spearman correlation -0.0964 ± 0.0322, while QGP and reconstruction have correlation 0.0355 ± 0.0188. Correlations over all sampled candidates are weak, but high-ranked QGP candidates are more often beneficial: precision is 0.600 in the top 1% and 0.569 in the top 5%. Consistent with this result, QGP produces substantially better finite patches than the matched FP-point, reconstruction, and random rankings across the five runs.

Large model evaluation-point replication. We repeat the matched evaluation-point test on Llama-2-7B using three prespecified fitting splits. Each arm uses the same ordinary affine Q_0, Q/K scope, group size 128, shared legal CPSR proposal pool, and held-out sets. The allocation is layer-balanced

at $K = 16$/layer over 32 layers, giving 512 relocations and 1024 changed coordinates per endpoint. Both arms use a single fixed ranking at this budget. All six endpoint patches are finite, legal, and have zero group-cardinality violations; the selected patches are unchanged when the per-module proposal pool is expanded from 4096 to 8192.

Table 9: Three-split Llama-2-7B matched Q-point versus FP-point replication. All values are ΔNLL relative to the same ordinary affine baseline for that split; negative is better.

Fit	Offset	Arm	ΔFit	ΔVal	ΔW2	ΔC4
0	0	Q-point	-0.1720	-0.0757	-0.0563	-0.0760
0	0	FP-point	+0.0227	+0.0080	+0.0180	+0.0199
17	34816	Q-point	-0.1079	-0.0330	-0.1236	+0.0250
17	34816	FP-point	+0.0043	-0.0016	-0.0030	-0.0016
33	67584	Q-point	-0.0351	-0.0593	-0.0247	+0.0381
33	67584	FP-point	+0.0019	-0.0024	+0.0030	+0.0146
Q-point mean ± std			-0.1050 ± 0.0685	-0.0560 ± 0.0215	-0.0682 ± 0.0505	-0.0043 ± 0.0625
FP-point mean ± std			$+0.0096 \pm 0.0114$	$+0.0013 \pm 0.0058$	$+0.0060 \pm 0.0108$	$+0.0110 \pm 0.0112$

Q-point improves fitting, validation, and W2 NLL in all three splits, whereas FP-point is near zero or slightly harmful on average. C4 is mixed: Q-point improves one of three splits and has mean ΔC4 close to zero.

A.3 Ordinary Affine Supplementary Evidence

Table 10 summarizes calibration robustness and allocation diagnostics with matched budgets. Each study uses its own matched protocol; deltas are relative to the corresponding baseline.

Table 10: Supplementary ordinary affine robustness and design diagnostics. Each study uses its own matched protocol; deltas are relative to the corresponding baseline.

Study	Setting	ΔVal	ΔW2	ΔC4
Calibration	W2, 25%	–	-0.0597	-0.0601
	W2, 50%	–	-0.0503	-0.0483
	W2, 100%	–	-0.0498	-0.0455
	C4, 100%	–	-0.0360	-0.0392
Allocation	Concentrated 6×64	-0.0748	-0.0714	-0.0828
	Global	-0.0709	-0.0627	-0.0853
	Uniform 24×16	**-0.1445**	**-0.1360**	**-0.0906**

All four calibration settings improve both held-out W2 and C4. The 25/50/100% labels denote the relative amount of calibration data used in this calibration study, rather than a fraction of the edit budget. The dashes in the calibration rows indicate fixed-budget diagnostics without an APG validation selection step. For Concentrated 6×64, the six layers are selected by the largest sum of their top-eight QGP candidate scores and 64 moves are taken from each selected layer; Uniform 24×16 instead distributes the same 384-move budget across all 24 layers. With the same 384-move budget, distributing edits across layers performs better than concentrating them in a small set of layers.

A separate Llama-2-7B control provides a large-model allocation check at the same 384-relocation budget. A global Q-point sort placed 25 relocations in layer 0 and 359 in layer 1, with none in the remaining 30 layers; it reduced the fitting loss but degraded validation and W2. Restoring the canonical layer-balanced allocation ($K = 12$ in each of 32 layers) changed only the allocation rule and recovered held-out improvement. The matched FP-point arm uses the same layer-balanced budget.

The layer-balanced result is consistent with the five-offset OPT finding that the deployed-state gradient provides the more useful fixed-budget ranking. The global sort shows the effect of allocation;

Table 11: Llama-2-7B Q-point/FP-point allocation control on one ordinary affine fitting instance. All rows use 384 relocations (768 changed coordinates) and preserve exact groupwise cardinality; negative ΔNLL is better.

Ranking / allocation	ΔFit	ΔVal	ΔW2	ΔC4
Q-point, global	-0.1101	+0.0869	+0.0824	-0.0907
FP-point, global	+0.0003	-0.0006	+0.0039	+0.0010
Q-point, 12/layer	**-0.1441**	**-0.0604**	**-0.0472**	**-0.0680**
FP-point, 12/layer	-0.0014	+0.0050	+0.0042	+0.0023

expanding the proposal pool from 4096 to 8192 per module leaves the selected layer-balanced patches unchanged.

A.4 Module Scope Dependence and Score Calibration

We additionally vary the eligible OPT-350M module class while holding the ordinary affine starting state, group size, five fitting offsets, and fixed 384-relocation (768-coordinate) budget constant. All reported patches are finite, legal, and preserve exact groupwise side-state cardinality.

Table 12: Matched module scope control on OPT-350M. Values are mean ± sample std ΔNLL over five fitting offsets; lower is better. Every row uses 384 relocations (768 changed coordinates).

Scope	ΔVal	ΔW2	ΔC4
Q/K	-0.2954 ± 0.0233	-0.3661 ± 0.0383	-0.1538 ± 0.0720
V/O	$+1.2891 \pm 0.3505$	$+1.2470 \pm 0.4313$	$+1.5465 \pm 0.3154$
FFN	-0.2963 ± 0.1276	-0.4738 ± 0.1650	-0.0626 ± 0.0955
All linear	$+1.1139 \pm 0.4038$	$+1.0210 \pm 0.4715$	$+1.3710 \pm 0.3368$

Q/K is consistently beneficial across the five offsets, and FFN also improves validation and W2 consistently while improving C4 on average. In contrast, V/O and naive all-linear pooling regress strongly under the same fixed edit budget. Useful relocations are not limited to Q/K, but the effect of a 384-move patch depends strongly on the module class.

We next inspect QGP score calibration across the six module classes by evaluating the top 128 candidates from each class individually on a held-out validation probe.

Table 13: QGP score calibration by module on OPT-350M. Score statistics use the full legal per-class pool; ρ and beneficial counts use the top 128 candidates evaluated individually.

Module	Median score	P99	Max	ρ(score,gain)	Beneficial / 128
Q	5.93×10^{-5}	7.11×10^{-4}	4.56×10^{-3}	0.3029	106
K	4.97×10^{-5}	7.50×10^{-4}	1.22×10^{-2}	0.4108	106
V	3.76×10^{-4}	1.20×10^{-2}	5.05×10^{-2}	0.2675	120
O	2.64×10^{-4}	1.24×10^{-3}	4.19×10^{-3}	0.2854	108
FC1	6.72×10^{-5}	1.65×10^{-3}	5.58×10^{-2}	0.0780	88
FC2	1.61×10^{-4}	1.06×10^{-3}	4.69×10^{-3}	0.0837	82

QGP score scales differ substantially across module classes. V has a larger median and upper tail than Q/K, while FC1 has the largest observed maximum despite a much weaker correlation between score and realized gain than K. A global sort therefore tends to favor high-scale V/FC1 candidates even though raw scores are not directly comparable across classes. For V/O, 120/128 individually probed V candidates and 108/128 O candidates are beneficial, but the selected 384-relocation patches contain 383–384 V moves per offset and degrade performance.

A.5 APG Trajectories and Large Model Profile

Table 14: OPT-350M adaptive-stopping trajectories.

Offset	K	Reloc.	ΔVal	ΔW2	ΔC4	Decision
0	16	384	-0.2890	-0.4077	-0.1469	select
	32	768	-0.2724	-0.4163	-0.1028	stop
	64	1536	-0.1809	-0.3778	+0.0266	diagnostic
2048	16	384	-0.3179	-0.4062	-0.1755	continue
	32	768	-0.3449	-0.4416	-0.1789	select
	64	1536	-0.3278	-0.4620	-0.1256	stop
4096	16	384	-0.3112	-0.4261	-0.1756	select
	32	768	-0.2901	-0.4485	-0.1415	stop
	64	1536	-0.2081	-0.4293	-0.0392	diagnostic

Table 15: Ordinary affine APG trajectories on large models.

Model	K/layer	Reloc.	Changed	ΔVal	ΔW2	ΔC4
Llama-2-7B	1	32	64	-0.0088	-0.0067	-0.0105
	2	64	128	-0.0176	-0.0138	-0.0170
	4	128	256	-0.0311	-0.0222	-0.0335
	8	256	512	-0.0528	-0.0373	-0.0625
	16	512	1024	-0.0776	-0.0493	-0.0950
	32	1024	2048	**-0.1043**	-0.0571	-0.1195
	64	2048	4096	-0.1026	-0.0724	-0.1657
Qwen3-8B	1	36	72	-0.0196	-0.0115	-0.0218
	2	72	144	-0.0726	-0.0399	-0.0537
	4	144	288	-0.1142	-0.0990	-0.0712
	8	288	576	**-0.1991**	-0.1571	-0.1406
	16	576	1152	-0.1972	-0.1671	-0.1506

For OPT-350M, APG selects $K^\star = 16, 32, 16$ across the three fitting offsets. For the ordinary affine large-model runs, APG selects $K^\star = 32$ on Llama-2-7B and $K^\star = 8$ on Qwen3-8B. Later W2/C4 prefixes are reported as post-selection diagnostics.

Table 16: Qwen3-8B A100 runtime profile. Timing is mean $\pm$ sample std over three trials after one warm-up.

Stage	Time (s)	Share
Gradient collection	2.173 ± 0.348	1.9%
QGP scoring	6.022 ± 0.848	5.2%
Compatibility selection	0.000186 ± 0.000026	$< 0.1\%$
APG validation forwards	107.836 ± 0.712	92.9%
Profiled pipeline	116.031 ± 0.615	100%

Table 17: Qwen3-8B PT^2 APG validation curve. Every point is finite, legal, and preserves exact groupwise side-state cardinality.

K/layer	Relocations	Changed	Validation NLL
1	36	72	3.26456 ± 0.00014
2	72	144	3.18953 ± 0.00034
4	144	288	3.12693 ± 0.00188
8	288	576	3.07895 ± 0.00277
16	576	1152	$\mathbf{3.06380 \pm 0.00389}$
32	1152	2304	3.10846 ± 0.00462

Peak allocated/reserved memory is 30.618/31.350 GiB. APG validation forwards account for 92.9% of the measured runtime. The resulting profile consists of one backward pass for ranking followed by several validation forwards. In all three stored Qwen fitting-split curves, $K = 16$ is both the minimum validation-loss prefix on the evaluated grid and the prefix immediately preceding the first non-improvement at $K = 32$.

A.6 REQUANT REFERENCE

ReQuant refinement-rule details. We include ReQuant as a post-quantization reference, although it uses different edit and cardinality constraints from CPSR. The configuration with matched calibration uses four reconstruction sweeps ($T = 4$), neighborhood parameter $K = 2$, strict negative quadratic reconstruction-delta acceptance, row-major traversal, and incremental gradient refresh.

Table 18: Separate ReQuant reference on the affine OPT-350M starting state. Negative ΔNLL is better.

Configuration	ΔVal	ΔW2	ΔC4	Changed	Card. viol.
ReQuant, matched calibration	-0.1086	-0.0480	-0.0186	2.66M	93,395
ReQuant, default budget	-0.0124	-0.0771	+0.0044	3.07M	93,520

Both variants remain exactly on the affine ternary codebook, but they change 2.66–3.07M coordinates and incur more than 93k group-cardinality violations.

A.7 STANDARDIZED DOWNSTREAM EVALUATION DETAILS

All downstream endpoints are evaluated after patch selection. For both models, HellaSwag, PIQA, ARC-Easy, and ARC-Challenge use the default `acc_norm` metric; WinoGrande and MMLU use `acc`. The macro score is the unweighted mean of these six displayed primary metrics. Batch size is 1 and maximum sequence length is 2048. Chance is 25% for HellaSwag and MMLU, 50% for PIQA and WinoGrande, and is computed empirically for ARC as the mean of $1/K_i$ over the evaluated examples, giving 25.016% for both ARC-Easy and ARC-Challenge under this task version.

Table 19: Per-split standardized zero-shot downstream results after PT^2, augmented with chance and full-precision references. FP uses the same primary metric convention as the ternary endpoints.

Endpoint	Hella.	PIQA	ARC-E	ARC-C	Wino.	MMLU	Macro
Chance reference	25.000	50.000	25.016	25.016	50.000	25.000	–
Llama-2-7B							
FP BF16 reference	75.991	79.271	74.621	46.331	69.061	40.778	64.342
PT^2 baseline	41.536	62.677	42.088	23.891	58.721	23.017	41.988
TernRefine-APG, fit 0	44.652	63.928	44.402	25.853	57.459	23.067	43.227
TernRefine-APG, fit 17	45.459	65.288	44.360	26.024	57.616	23.394	43.690
TernRefine-APG, fit 33	44.324	63.983	44.318	25.597	56.196	23.437	42.976
Qwen3-8B							
FP BF16 reference	74.935	77.693	80.682	56.741	68.114	72.945	71.852
PT^2 baseline	30.323	56.202	33.460	23.379	51.776	22.903	36.340
TernRefine-APG, fit 0	31.75	56.09	36.99	24.40	53.12	22.91	37.54
TernRefine-APG, fit 17	32.58	56.37	37.92	25.34	52.80	23.00	38.00
TernRefine-APG, fit 33	32.25	57.24	37.46	25.00	51.07	22.94	37.66

The displayed macro average increases for all three fitted endpoints on both models, while the absolute gap to full precision remains substantial and the per-task changes are heterogeneous. MMLU remains near the 25% chance reference, ARC-Challenge stays near its empirical chance level, and Llama WinoGrande decreases. The reported standard deviations quantify variation across the three independently fitted patches.

For Llama-2-7B, the original evaluation retained per-example samples for the PT^2 baseline and all three fitted endpoints, enabling a paired example-level bootstrap. We compare PT^2 correctness to the per-example mean correctness of fits 0/17/33 and use 10,000 paired resamples. Macro resampling is stratified by task so that the six tasks remain equally weighted.

Table 20: Llama-2-7B paired downstream analysis from retained per-example logs. Δ is the mean of the three TernRefine fits minus PT^2 in percentage points.

Task	N	Δ (pp)	Paired bootstrap 95% CI (pp)
HellaSwag	10042	+3.276	$[+2.865, +3.704]$
PIQA	1838	+1.723	$[+0.725, +2.702]$
ARC-Easy	2376	+2.273	$[+1.375, +3.213]$
ARC-Challenge	1172	+1.934	$[+0.796, +3.072]$
WinoGrande	1267	-1.631	$[-3.210, -0.053]$
MMLU	14015	+0.278	$[+0.059, +0.492]$
Macro	–	+1.309	$[+0.895, +1.719]$

The paired macro shift is positive, while WinoGrande is negative and MMLU shows a small positive shift from a near-chance baseline. The sample-level MMLU analysis, formed by concatenating the retained subject-level records, gives a PT^2 point estimate of 23.040%, slightly different from the harness group aggregate 23.017% retained in Table 19; the primary table uses the harness aggregate. Qwen's original downstream run retained aggregate metrics but not per-example samples, so its uncertainty summary uses across-fit variation rather than a paired bootstrap.

A.8 PT^2 Replication and Residual Diagnostics

We report the full PT^2 replication results here to keep the main text comparison compact. The PT^2-LLM Llama-2-7B deployment uses group size 128, 128 calibration samples, sequence length 2048, and the official ATQ+SSR path. It covers all 32 decoder layers and all 224 ternary modules, including the 64 Q/K modules used by TernRefine; it passes state parity and detached reload and preserves all 64 SSR permutations as bijections. The Qwen3-8B deployment covers all 36 decoder layers and 72 Q/K ternary modules; its reported refined states remain finite and legal, SSR permutations

remain bijective, and every selected patch preserves exact groupwise side-state cardinality. The strict PT^2 state-parity gate was not run for this Qwen endpoint; all other reported legality, finiteness, permutation, and cardinality checks passed. The fitting splits are 0/17/33 and the fixed selection split uses disjoint indices 1–16.

Table 21: Full per-split language modeling results after PT^2 initialization, with full-precision and PT^2 anchors (provenance family: `LM-robustness`). ΔNLL for fitted rows is relative to the frozen PT^2 baseline; negative indicates improvement.

Endpoint	$K^\star$	Changed	W2 PPL	C4 PPL	ΔW2	ΔC4
Llama-2-7B						
FP BF16 reference	–	–	5.4729	7.2666	–	–
PT^2 baseline	–	–	11.6430	24.3244	–	–
Fit 0	8	512	10.9103	22.3797	-0.0650	-0.0833
Fit 17	16	1024	10.6336	20.8326	-0.0907	-0.1550
Fit 33	16	1024	10.7866	22.0230	-0.0764	-0.0994
TR mean	–	–	10.7769 ± 0.1386	21.7451 ± 0.8101	$\mathbf{-0.0774 \pm 0.0129}$	$\mathbf{-0.1126 \pm 0.0376}$
Qwen3-8B						
FP BF16 reference	–	–	9.7252	15.4290	–	–
PT^2 baseline	–	–	49.204	175.976	–	–
Fit 0	16	1152	33.282	109.299	-0.3909	-0.4763
Fit 17	16	1152	32.432	105.472	-0.4168	-0.5119
Fit 33	16	1152	32.819	107.552	-0.4050	-0.4924
TR mean	–	–	32.844 ± 0.426	107.441 ± 1.916	$\mathbf{-0.4043 \pm 0.0130}$	$\mathbf{-0.4935 \pm 0.0178}$

All three fitted patches improve both W2 and C4 on both large models. Using NLL for linear gap accounting, the mean refined endpoints recover 10.25% of the FP-to-PT^2 W2 gap and 9.32% of the C4 gap on Llama-2-7B; the corresponding Qwen3-8B recoveries are 24.93% and 20.27%.

Endpoint provenance. The language modeling and downstream studies use independently materialized TernRefine endpoint families from the same frozen PT^2 checkpoint and protocol, labeled `LM-robustness` and `DS-eval`, respectively. Table 21 reports `LM-robustness`; the standardized downstream table and the held-out NLL table below report `DS-eval`. All `DS-eval` endpoints pass the reported finiteness, legality, cardinality, and edit-count checks, and use disjoint fitting and selection splits.

Table 22: Held-out NLL for the Llama-2-7B endpoints used in standardized downstream evaluation (provenance family: `DS-eval`).

Fit	$K^\star$	Reloc.	Changed	ΔW2	ΔC4
0	8	256	512	-0.0907	-0.1431
17	16	512	1024	-0.1138	-0.1938
33	16	512	1024	-0.0494	-0.0838
Mean	–	–	–	$\mathbf{-0.0846 \pm 0.0327}$	$\mathbf{-0.1403 \pm 0.0551}$

Table 23: Post-selection single-relocation residual diagnostics on untouched W2 for Llama-2-7B.

Initializer	Sample	$\rho(S, -\Delta L_{\mathrm{W2}})$	Mean ΔL_{W2}	Beneficial density
Ordinary affine	Ranked	0.1284	-0.000505	0.6836
	Random	–	-0.000361	0.6563
PT^2 ATQ+SSR	Ranked	0.1109	+0.000513	0.3242
	Random	–	+0.000616	0.3125

The beneficial density of ranked W2 relocations decreases from 68.36% under the ordinary affine initializer to 32.42% after PT^2 on Llama-2-7B. After PT^2, beneficial ranked moves are less frequent, but TernRefine still improves both W2 and C4 in all three fitting splits.