Route-Consistent Adaptation for Stable Quantization of Mixture-of-Experts Models with Theoretical Guarantees

Longteng Zhang[1], **Sen Wu**[3], **Qiang Wang**[3*], **Shaohuai Shi**[3], **Xiaowen Chu**[1,2*]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Harbin Institute of Technology, Shenzhen
lzhang330@connect.hkust-gz.edu.cn, 24S151068@stu.hit.edu.cn
qiang.wang@hit.edu.cn, shaohuais@hit.edu.cn, xwchu@ust.hk

Abstract

Quantizing Mixture-of-Experts (MoE) models is difficult for two coupled reasons: expert quantization error perturbs expert outputs, and sparse routing amplifies small perturbations into discrete expert selection changes. This coupling produces progressive output drift that is not captured by dense model quantization analyses. To address this gap, we propose a unified quantization framework that jointly handles both sources of error. First, we formulate compression-induced routing inconsistency as a form of training-inference router mismatch and use this connection to motivate limited routing replay, which replays BF16 routes during quantization-aware fine-tuning (QAT), freezes router parameters, and updates only expert-side adapter parameters. We then provide a standard convergence guarantee for the resulting fixed-route QAT objective, together with a routing-consistency bound under explicit router-margin conditions. Second, for expert restoration under a fixed rank budget, we propose load-aware low-rank compensation that assigns larger compensation rank to heavily loaded experts. Under this allocation rule, we prove monotone strict decrease of weighted compression error and characterize optimal rank assignment by marginal load-weighted spectral gains. Finally, experiments on the MXFP4 quantization of Mixtral, DeepSeek, and Qwen3 MoE models show that our method improves perplexity and downstream accuracy over prior methods. We also report system efficiency on Qwen3-30B, where uniform W4A8 serving accelerates decoding and our adapter path preserves most of that acceleration.

1 Introduction

Mixture-of-Experts (MoE) architectures scale model capacity by activating only a small subset of experts for each token, making them a central design for modern large language models Yang et al. [2024a,b, 2025], DeepSeek-AI [2025], OpenAI [2025]. This sparse conditional computation improves the capacity-compute trade-off, but does not remove the deployment bottleneck: MoE models still contain a very large number of parameters, and reducing their memory and serving cost remains essential Ma et al. [2025a]. Consequently, quantization and compression are natural tools for MoE deployment. However, MoE is harder to compress than dense transformers because compression affects not only continuous hidden-state computation, but also the discrete routing decisions that determine which experts are executed Chen et al. [2025], Fu et al. [2026], Fang and Huang [2025], Chowdhury et al. [2026].

*Corresponding author

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

This routing dependence, in turn, makes MoE compression uniquely fragile. In dense models, quantization noise mainly perturbs layer outputs smoothly Frantar et al. [2023], Lin et al. [2024], Liu et al. [2025a], Li et al. [2024]. In MoE models, by contrast, a small perturbation to router scores can change the top-k expert set, reroute the token through different subnetworks, and then alter the weighted aggregation path in downstream layers. Recent evidence shows that router errors dominate much of the performance loss in low-bit MoE inference, and that the most common failure mode is rank flipping near the routing boundary rather than arbitrary corruption of the full score distribution Fang and Huang [2025], Chen et al. [2025]. At the same time, expert sensitivity is not uniform: some experts are more critical to model quality than others, which makes uniform compression unnecessarily wasteful Chowdhury et al. [2026].

Together, these observations have motivated two largely separate lines of work. One line studies routing-aware quantization, seeking to preserve expert selection under low precision Fang and Huang [2025]. Another line studies expert-sensitive compression, allocating restoration or precision according to expert importance Chowdhury et al. [2026], Li et al. [2025], Kim et al. [2023]. In parallel, low-rank residual compensation has emerged as an effective way to recover information lost by pruning or approximation while remaining system-efficient when fused with the main computation Zhang et al. [2026], Huang et al. [2025], Liu et al. [2025b]. However, existing approaches still tend to treat routing stabilization and expert restoration as separate problems. This separation is limiting in MoE, because the experts that matter most after compression are precisely those that receive the routed probability mass, and that mass itself depends on how stable the router remains under compression. A related lesson comes from recent studies of MoE reinforcement learning: even when model parameters are unchanged, a mismatch between training-time and inference-time routing can destabilize optimization and degrade final performance Ma et al. [2025b], Shao et al. [2024]. Although that setting is different from quantization, it reveals a broader principle: in MoE, route consistency is not a secondary detail but part of the optimization target. For compression, this suggests that preserving routing during fine-tuning is likely to be as important as reducing local weight reconstruction error. It also suggests that restoration capacity should be allocated according to the routed load, rather than uniformly between experts.

These trends point to a clear opportunity. An effective MoE compression method should jointly address three requirements: it should preserve routing decisions near the top-k boundary, restore the computation of the most heavily used experts, and remain compatible with practical low-precision deployment. In particular, relying on heterogeneous expert-wise bit formats can improve accuracy but often complicates kernel execution and weakens system efficiency Chowdhury et al. [2026], Chitty-Venkata et al. [2025], NVIDIA [2024], Yan and Hongxiao Bai [2026]. A more practical approach is to keep a hardware-friendly base format and use lightweight adaptation only where compression creates the largest functional error.

Given these requirements, we propose a unified quantization framework for MoE that couples routing stabilization with expert restoration. Our method combines limited routing replay during fine-tuning to suppress compression-induced route drift, and load-aware low-rank compensation to concentrate restoration capacity on experts that carry the largest routed mass. This design directly targets the two main sources of MoE compression error, i.e. discrete routing instability and uneven expert sensitivity, within a single optimization process. We provide theoretical guarantees and an end-to-end practical recipe with MXFP4 quantization, showing that MoE compression can be both accurate and deployment-friendly when routing and restoration are treated jointly rather than in isolation.

2 Related Work

MoE quantization. Recent MoE PTQ methods show that preserving router behavior is critical. Rank-aware PTQ methods reduce top-k route flips via calibration objectives, but usually rely on intricate objective balancing and still treat routing and expert restoration as partially decoupled Fang and Huang [2025]. Dedicated MoE quantization benchmarks such as QuantMoE-Bench emphasize that expert and router sensitivity vary sharply across models and layers Li et al. [2025]. MoQE studies low-bit expert quantization as a precursor to this line of work Kim et al. [2023]. Automated fine-grained MoE quantization and theory-driven efficient MoE quantization allocate precision according to structural sensitivity or generalization-motivated expert importance Xie et al. [2025], Chowdhury et al. [2026]. Mixed-precision expert methods such as MoPEQ improve quality by assigning heterogeneous bit widths, but the resulting format diversity can complicate runtime optimization on commodity

acceleration stacks Chitty-Venkata et al. [2025]. In contrast, our method keeps a uniform MXFP4 expert format and uses low-rank compensation to recover accuracy under a matched memory budget.

Router-consistency and replay. A second thread focuses on stabilizing the router itself. Work on MoE RL stabilization identifies training-inference router mismatch as a source of instability and introduces routing replay to align policies Ma et al. [2025b]. Other recent work studies uncertainty-aware mechanisms for stabilizing quantized MoE routing Ordóñez et al. [2026]. Our work transfers the replay principle to quantization-aware compression and makes the route-cache protocol explicit for adapter-only MXFP4 fine-tuning.

Compression compensation. Finally, low-rank residual compensation and sparsity-aware adaptation show that truncated-SVD-style residual modeling can provide provable reconstruction gains Zhang et al. [2026]. MiLo and bandwidth-efficient adaptive MoE compensation are especially close because they also use low-rank compensators for quantized MoE experts Huang et al. [2025], Liu et al. [2025b]. Our distinction is that rank assignment is tied to cached routed load and is trained under fixed BF16 route replay, so the compensation budget is allocated to experts that are both spectrally sensitive and frequently selected.

3 Problem Formulation

3.1 Preliminary

We first set notation for a single MoE block with M experts and top-k routing. For token representation $x \in \mathbb{R}^d$, router logits are

$$\boldsymbol{s}(x) = g_{\theta_r}(x) \in \mathbb{R}^M. \tag{1}$$

Let $\mathcal{I}(x) = \mathrm{TopK}(\boldsymbol{s}(x), k) \subseteq \{1, \dots, M\}$ and router weights

$$\pi_j(x) = \frac{\exp(s_j(x))}{\sum_{\ell \in \mathcal{I}(x)} \exp(s_\ell(x))} \mathbf{1}\{j \in \mathcal{I}(x)\}. \tag{2}$$

The block output is

$$y(x) = \sum_{j=1}^{M} \pi_j(x) f_j(x; W_j), \tag{3}$$

where f_j is expert j with parameters W_j.

Let $Q(\cdot)$ denote quantization (our work focuses on MXFP4 formats motivated by emerging micro-scaling hardware support NVIDIA [2024, 2025]) and $\widehat{W}_j = Q(W_j)$. Quantized experts are $\widehat{f}_j(x) = f_j(x; \widehat{W}_j)$, and quantized router logits are $\widehat{\boldsymbol{s}}(x)$ with selected set $\widehat{\mathcal{I}}(x)$, when activations and weights in the router are quantized or perturbed through preceding blocks. We define Δ_{exp} and Δ_{route} as the expert reconstruction error conditioned on the BF16 route and the route mismatch probability, respectively.

Definition 1 (Quantization-induced MoE distortion)**.** *For distribution $\mathcal{D}$ over tokens, define*

$$\Delta_{\mathrm{exp}} = \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{j \in \mathcal{I}(x)} \pi_j(x) \| f_j(x) - \widehat{f}_j(x) \|_2 \right], \tag{4}$$

$$\Delta_{\mathrm{route}} = \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbf{1}\{\mathcal{I}(x) \neq \widehat{\mathcal{I}}(x)\} \right]. \tag{5}$$

Given these two error terms, assume each expert output is bounded by $\|f_j(x)\|_2, \|\widehat{f}_j(x)\|_2 \leq B_f$. By adding and subtracting the quantized output evaluated on the BF16 route and applying the triangle inequality, we obtain

$$\mathbb{E}\|y(x) - \widehat{y}(x)\|_2 \leq \Delta_{\mathrm{exp}} + 2B_f \, \Delta_{\mathrm{route}} + B_f \, \mathbb{E}\|\pi(x) - \widehat{\pi}(x)\|_1. \tag{6}$$

This explicitly shows why controlling route mismatch is as important as reducing local quantization error.

3.2 Problem Formulation

We now formulate MoE compression as minimizing a coupled objective under a low-precision and compensation budget:

$$\min_{\widehat{W},\,\{r_j\}} \; \mathcal{L}_{\text{task}}(\widehat{W}) + \lambda\,\mathcal{R}_{\text{route}}(\widehat{W}) \quad \text{s.t.} \quad \sum_{j=1}^{M} r_j \le R,\; \widehat{W} \in Q(\cdot),\; r_j \in \mathbb{Z}_{\ge 0}. \tag{7}$$

Here r_j is compensation rank for expert j, and $\mathcal{R}_{\text{route}}$ penalizes routing inconsistency.

To make the route-optimization connection explicit, define two route policies over experts for the same input x:

$$\pi_{\text{inf}}(\cdot \mid x) \propto \exp(\boldsymbol{s}^{\text{bf16}}(x)), \qquad \pi_{\text{tr}}(\cdot \mid x) \propto \exp(\boldsymbol{s}^{\text{mxfp4}}(x)). \tag{8}$$

These policies correspond to inference-training bias in RL and uncompressed-compressed routing in quantization, respectively. Thus, we can connect quantization-induced routing inconsistency to the training-inference router mismatch studied in MoE RL.

Given this shared objective, we now state Proposition 3.1, which formalizes that compression-induced route drift belongs to the same instability class as rollout inconsistency and that fixed-route replay directly removes this instability during adapter fine-tuning.

Proposition 3.1. *Suppose both settings share identical token distribution $\mathcal{D}$ and top-k selection rule, and fix a logit gauge by subtracting the mean logit from each vector. Assume the centered router logits are bounded and that the BF16 route margin*

$$\gamma(x) = s^{\text{inf}}_{(k)}(x) - s^{\text{inf}}_{(k+1)}(x) \tag{9}$$

is positive on the tokens of interest. If

$$\|\boldsymbol{s}^{\text{inf}}(x) - \boldsymbol{s}^{\text{tr}}(x)\|_\infty < \gamma(x)/2, \tag{10}$$

then the top-k route is unchanged. Consequently, any objective that controls centered logit drift, or controls policy drift together with an explicit inverse-stability condition on the softmax map, upper-bounds quantization route mismatch on large-margin tokens. In this sense, policy inconsistency is a useful surrogate for route inconsistency, even though not an unconstrained optimization-order equivalent objective.

Proof. See Appendix A.1. □

4 Methodology

Building on the formulation above, this section develops the two components of our method. We begin by introducing limited replay and its theoretical guarantees for convergence and routing consistency during quantization-aware fine-tuning (QAT). Next, we propose the load-aware expert compression module to reduce the cumulative information loss brought by expert quantization. Finally, we summarize the full pipeline that couples limited routing replay and load-aware compensation in Algorithm 1.

4.1 Limited Routing Replay for Quantization-Aware Fine-Tuning

Operationally, we sample a replay set $\mathcal{D}_{\text{rep}} = \{(x_t, y_t)\}_{t=1}^{N}$ from the fine-tuning corpus and run the BF16 model once over the full replay set. During this pass, we cache token-aligned route tuples and simultaneously record expert workload:

$$\mathcal{C} = \{(x_t, y_t, \mathcal{I}_t^\star, \boldsymbol{\pi}_t^\star)\}_{t=1}^{N}, \tag{11}$$

where $\mathcal{I}_t^\star = \text{TopK}(g_{\theta_r}(x_t), k)$ and $\boldsymbol{\pi}_t^\star$ are BF16 route weights. QAT minibatches are then sampled only from this cached replay set so every token has a token-aligned BF16 route. Meanwhile, expert workload is measured from the same cached routes and normalized by the number of routed expert assignments:

$$p_j = \frac{1}{Nk} \sum_{t=1}^{N} \mathbf{1}\{j \in \mathcal{I}_t^\star\}, \qquad \sum_{j=1}^{M} p_j = 1. \tag{12}$$

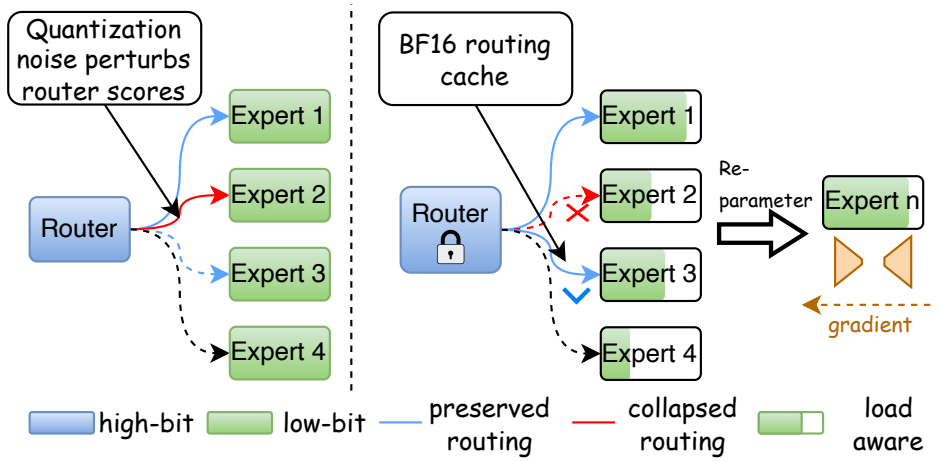


Figure 1: **Illustration of Limited Routing Replay and Load-Aware Compensation.** Quantization noise perturbs router scores and can change the top-k expert set, causing low-bit MoE inference to deviate from the original routing path. Our method caches token-aligned BF16 routes and replays them during QAT with the router frozen, so gradients update only expert-side low-rank compensation adapters. Compensation rank is allocated according to cached expert load, giving heavily routed experts more restoration capacity while preserving a hardware-friendly low-bit serving path.

All parameters except the low-rank compensation adapter (introduced in Section 4.2) are kept frozen during QAT, and for each token, expert selection is replayed from $\mathcal{C}$. We call this *limited replay* because replay is enforced on sparse route decisions while gradients still propagate through the task loss to adapter parameters.

We then derive the replayed loss:

$$F(\phi) = \frac{1}{N}\sum_{t=1}^{N} \ell\left(\sum_{j\in\mathcal{I}_t^\star} \pi_{t,j}^\star \widehat{f}_j(x_t;\phi_j),\, y_t\right), \tag{13}$$

where ϕ denotes only trainable low-rank adapter parameters under frozen quantized base weights. If F is L-smooth and satisfies the PL inequality with constant $\mu > 0$ on the reachable region:

$$\frac{1}{2}\|\nabla F(\phi)\|_2^2 \geq \mu\big(F(\phi) - F^\star\big), \tag{14}$$

then the following two results connect replayed optimization to stable routing. Theorem 4.1 gives convergence of the fixed-route adapter objective, and Theorem 4.2 states the corresponding routing-consistency guarantee.

Theorem 4.1. *Under the assumption above, gradient descent*

$$\phi^{t+1} = \phi^t - \eta\nabla F(\phi^t), \qquad 0 < \eta \leq \frac{1}{L}, \tag{15}$$

converges linearly:

$$F(\phi^t) - F^\star \leq (1-\eta\mu)^t\big(F(\phi^0) - F^\star\big). \tag{16}$$

Proof. See Appendix A.2. □

Theorem 4.2. *On replayed tokens $\{x_t\}$, limited replay enforces exact routing consistency:*

$$\mathcal{I}_{\text{train}}(x_t) = \mathcal{I}_t^\star, \quad \boldsymbol{\pi}_{\text{train}}(x_t) = \boldsymbol{\pi}_t^\star \quad \forall t. \tag{17}$$

If in deployment router logits satisfy

$$\|\widehat{\boldsymbol{s}}(x) - \boldsymbol{s}^\star(x)\|_\infty \leq \varepsilon \tag{18}$$

with probability at least $1-\delta$, and route margin $\gamma(x) \geq 2\varepsilon$ with probability at least $1-\rho$, then

$$\Pr\left[\widehat{\mathcal{I}}(x) = \mathcal{I}^\star(x)\right] \geq 1 - (\delta + \rho). \tag{19}$$

Proof. See Appendix A.3. □

Compared with calibration-only rank-alignment losses that must learn route stability indirectly, limited replay removes route non-stationarity during compression fine-tuning. This isolates expert restoration as the main optimization target and yields stable gradients with explicit convergence path.

4.2 Load-Aware Expert Compression Compensation

We now introduce expert restoration. After quantization, each expert has residual matrix

$$E_j = W_j - Q(W_j). \tag{20}$$

We approximate E_j by rank-r_j truncated SVD compensation E_{j,r_j}. Let singular values of E_j be $\sigma_{j,1} \geq \sigma_{j,2} \geq \cdots \geq 0$. Then per-expert residual error is

$$\varepsilon_j(r_j) = \|E_j - E_{j,r_j}\|_F^2 = \sum_{m>r_j} \sigma_{j,m}^2. \tag{21}$$

Using the one-shot cached workload p_j, we allocate ranks before fine-tuning. Specifically, we minimize weighted error under budget R:

$$\min_{\{r_j\}} \ \mathcal{E}(\mathbf{r}) = \sum_{j=1}^{M} p_j \varepsilon_j(r_j) \quad \text{s.t.} \quad \sum_{j=1}^{M} r_j \leq R,\ r_j \in \mathbb{Z}_{\geq 0}. \tag{22}$$

This objective naturally yields a marginal-gain view. Define the gain of assigning one extra rank to expert j at current r_j:

$$\Delta_j(r_j) = p_j\big(\varepsilon_j(r_j) - \varepsilon_j(r_j+1)\big) = p_j \sigma_{j,r_j+1}^2. \tag{23}$$

The load-aware allocation greedily assigns each next rank unit to the expert with maximal $\Delta_j(r_j)$, which makes the rank allocation problem especially transparent. Since adding one rank to expert j removes exactly the load-weighted spectral energy $p_j\sigma_{j,r_j+1}^2$, the weighted compression error decreases by the selected marginal gain at every step. Moreover, the full budgeted problem can be viewed as selecting R unit-rank gains from all experts. Thus, greedy load-aware allocation is not only intuitively aligned with routed workload, but also admits a simple optimality characterization. We state these properties below.

Theorem 4.3. *Assume for every allocation step $t < R$, there exists some expert j with $\sigma_{j,r_j^{(t)}+1} > 0$. Let $\mathbf{r}^{(t)}$ be greedy load-aware allocation after t rank units. Then*

$$\mathcal{E}(\mathbf{r}^{(t+1)}) < \mathcal{E}(\mathbf{r}^{(t)}) \quad \forall t < R. \tag{24}$$

Hence weighted compression error is strictly decreasing with budget.

Proof. See Appendix 4.3. □

Proposition 4.4. *For any fixed budget R, an allocation $\mathbf{r}^\star$ is globally optimal iff the selected set of R unit gains*

$$\{p_j\sigma_{j,m}^2 : 1 \leq m \leq r_j^\star\} \tag{25}$$

contains the R largest values among all unit gains $\{p_j\sigma_{j,m}^2\}_{j,m}$.

Proof. See Appendix 4.4. □

This yields a direct interpretation: heavily loaded experts with large remaining singular values receive higher ranks first; lightly loaded experts naturally receive low or zero compensation when their load-weighted gains are small.

5 Experimental Results

In this section, we evaluate MXFP4 quantization on three representative MoE LLMs. For each model, we compare our method against strong baselines that target either routing stabilization or expert restoration, and we perform ablations to isolate the contributions of replay and load-aware compensation. We also measure route mismatch rates and system throughput to verify that our method achieves better accuracy without sacrificing MXFP4 efficiency.

Models. We conduct QAT on three representative MoE LLMs: Mixtral-8×7B Jiang et al. [2024], DeepSeek-v2-16B DeepSeek-AI [2025], and Qwen3-30B Yang et al. [2025].

Datasets. Following prior work Wang et al. [2024a,b], Zhang et al. [2026], we evaluate on datasets matched to different task types. We use MetaMathYu et al. [2023] for the MATH domain, and auxiliary multiple-choice training questions from ARC, MC-TEST, OBQA, RACE, etc., for the multidisciplinary domain.

Baselines. We compare our method with Uniform MXFP4 PTQ, which quantizes per-channel weight without route stabilization and error compensation. Together, we evaluate two recent MoE specialized quantization methods that involve Rank-Aware PTQ Fang and Huang [2025] which aligns top-k during calibration and Mixed-Precision PTQ Chowdhury et al. [2026] which assigns expert-wise heterogeneous bit-width.

Metrics. We report perplexity (C4, WikiText-2), average 5-shot accuracy (MMLU Hendrycks et al. [2021], GSM8K Cobbe et al. [2021], ARC-C Talmor et al. [2019]), route mismatch rate (Δ_{route}), and system throughput (tokens/s) on RTX PRO 6000 Blackwell GPU.

Implementation details. Replay cache and expert workload are collected jointly from a one-shot BF16 pass over the replay set with top-k routes. Unless stated otherwise, we use 4096 replay sequences sampled from the fine-tuning corpus, sequence length 2048, AdamW with learning rate $2e^{-5}$, cosine decay, batch size 16, and 3 epochs. Rank allocation is performed before QAT with average rank 128 per expert ($R = 128M$). During QAT, only low-rank adapters in expert feed-forward projections are fine-tuned; router parameters, attention layers, embeddings, and base MXFP4 expert weights remain frozen. Δ_{route} is measured per MoE layer as the fraction of evaluation tokens whose compressed top-k expert set differs from the BF16 top-k set, then averaged across MoE layers and evaluation samples.

Budget matching. Mixed-precision baselines are constrained to the same total expert-parameter memory as uniform MXFP4 plus adapters, including scale metadata and adapter weights. We report the main end-to-end comparison in Table 1.

Table 1: End-to-end MXFP4 compression results. Lower perplexity is better; higher accuracy is better.

Model	Method	C4 ppl↓	Wikitext2 ppl↓	Avg Acc.↑	Δ_{route} ↓
Mixtral-8×7B	Uniform MXFP4 PTQ	11.92	7.31	56.4	18.7%
	Rank-Aware PTQ	11.21	6.98	58.1	9.2%
	Mixed-Precision MoE	10.98	6.87	58.9	5.9%
	Ours (LRR+LAC)	**10.41**	**6.55**	**62.8**	**1.3%**
DeepSeek-v2-16B	Uniform MXFP4 PTQ	10.87	6.74	61.0	11.9%
	Rank-Aware PTQ	10.36	6.48	62.2	8.8%
	Mixed-Precision MoE	10.14	6.39	62.6	4.5%
	Ours (LRR+LAC)	**9.71**	**6.10**	**64.1**	**1.1%**
Qwen3-30B	Uniform MXFP4 PTQ	9.93	6.21	74.8	7.1%
	Rank-Aware PTQ	9.55	6.03	75.7	4.1%
	Mixed-Precision MoE	9.44	5.95	76.1	3.8%
	Ours (LRR+LAC)	**9.06**	**5.71**	**78.6**	**0.9%**

5.1 End-to-End Comparison

The results in Table 1 show that our method consistently yields lower perplexity and higher downstream accuracy, while reducing route mismatch by roughly $4\times$-$8\times$ versus strong PTQ baselines.

Specifically, the route mismatch results show that accuracy recovery is strongly tied to routing stability. Uniform MXFP4 PTQ suffers substantial route drift, especially on Mixtral-8×7B, where 18.7% of routed token decisions differ from the BF16 model. Rank-Aware PTQ reduces this mismatch by explicitly aligning top-k routes during calibration, but still leaves nontrivial instability after quantization. Mixed-Precision MoE further improves quality by allocating higher precision to sensitive experts, yet its route mismatch remains several times larger than ours. In contrast, LRR+LAC reduces the average mismatch rate from 12.6% under Uniform MXFP4 PTQ to 1.1%, corresponding to an 11.4× reduction. Even compared with the strongest PTQ baseline, Mixed-Precision MoE, our mismatch rate is 4.3× lower.

5.2 Ablation Studies

We now isolate the effect of each component in LRR+LAC. We first study whether limited routing replay is responsible for route stabilization, then examine whether the proposed load-aware rank allocation improves over simpler allocation rules under the same total rank budget. Finally, we analyze route-margin sensitivity and spectrum-level preservation to better understand where the gains come from.

Effect of routing replay. The results in Table 2 show that replay sharply improves route consistency and explains most of the accuracy recovery; compensation further improves language modeling quality by reducing expert reconstruction error on heavily used experts.

Table 2: Ablation on Mixtral-8×7B MXFP4.

Variant	C4 ppl↓	Avg Acc.↑	Δ_{route} ↓
No replay, no compensation	11.92	56.4	18.7%
Replay only (LRR)	10.83	61.6	1.4%
Compensation only (uniform rank)	11.08	58.7	17.1%
Replay + uniform compensation	10.63	62.0	1.4%
Replay + load-aware compensation (full)	**10.41**	**62.8**	**1.3%**

Rank allocation policy. We next examine whether the rank-allocation rule matters once replay is fixed. Under fixed total rank budget $R = 128$ (all experts, all MoE layers), we compare allocation strategies. The best results in Table 3 align with the theorem-derived marginal gain criterion, confirming that both load and spectrum matter.

Table 3: Compensation rank allocation strategies on DeepSeek-v2-16B.

Allocation strategy	Weighted error $\mathcal{E}$ ↓	C4 ppl↓	Avg Acc.↑
Uniform per expert	1.000 (norm.)	10.04	63.1
Load-only proportional ($r_j \propto p_j$)	0.934	9.91	63.5
Spectral-only greedy ($\sigma^2_{j,m}$)	0.918	9.88	63.6
Load-aware spectral greedy ($p_j\sigma^2_{j,m}$)	**0.873**	**9.71**	**64.1**

Margin and route stability. To connect the empirical behavior back to the margin condition in Theorem 4.2, we partition tokens by BF16 route margin $\gamma = s_{(k)} - s_{(k+1)}$. Without replay, low-margin buckets dominate route flips; replay removes this sensitivity during fine-tuning and reduces post-tuning instability. For example, for $\gamma \in [0, 0.05]$, mismatch drops from 19.4% to 3.1%; for $\gamma > 0.2$, both are already below 1%.

Spectrum-level information preservation. Finally, to directly test the claim that our method preserves more useful information under MXFP4 compression, we analyze expert-weight singular-value energy distributions against the baselines. We use BF16 as reference and report (i) distribution-level distances (L1 and Jensen-Shannon divergence), (ii) top-k cumulative energy retention ($E@k$), and (iii) structural spectrum.

Specifically, for each method m, let singular values be $\{\sigma_{m,i}\}_{i=1}^n$ and normalized spectral energy be

$$e_{m,i} = \frac{\sigma^2_{m,i}}{\sum_{j=1}^n \sigma^2_{m,j}}, \qquad \sum_{i=1}^n e_{m,i} = 1. \tag{26}$$

We partition sorted components into contiguous index bands $\{B_b\}_{b=1}^{B}$ and compute relative band-energy error to BF16:

$$R_{m,b} = 100 \cdot \frac{\sum_{i \in B_b} e_{m,i} - \sum_{i \in B_b} e_{\mathrm{BF16},i}}{\sum_{i \in B_b} e_{\mathrm{BF16},i}}. \tag{27}$$

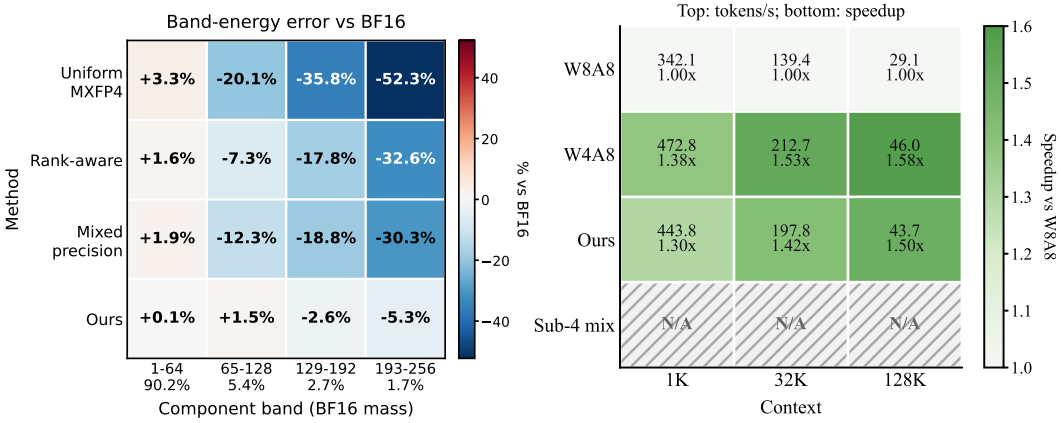
The preservation score:

$$\mathrm{Score}(m) = 1 - \frac{1}{2} \left\| \mathbf{e}_m - \mathbf{e}_{\mathrm{BF16}} \right\|_1 . \tag{28}$$

We report this score in Table 4, where higher score means closer global spectral shape to BF16. The spectral-error results demonstrate that although the preservation-score gap to the best baseline is only 0.0176, the equivalent L1 distance drops from 0.0932 to 0.0581, i.e., a 37.7% relative reduction in spectral error.

Table 4: Spectrum-preservation metrics versus BF16 reference. Lower is better for L1/JSD; higher is better for Preservation Score and $E@k$.

Method	Preservation Score↑	L1↓	JSD↓	$E@64$	$E@128$
Uniform MXFP4 PTQ	0.9480	0.1040	0.0029	0.9313	0.9746
Rank-Aware PTQ	0.9534	0.0932	0.0019	0.9162	0.9664
Mixed-Precision MoE	0.9392	0.1217	0.0023	0.9188	0.9663
Ours (LRR+LAC)	**0.9710**	**0.0581**	**0.0006**	0.9026	0.9577



(a) (b)

Figure 2: (a) Relative band-energy error versus BF16. Each cell reports $R_{m,b}$, the percentage change in a method's normalized band energy relative to the BF16 energy mass in the same band. Positive values indicate excess energy concentration in that band; negative values indicate a deficit. We use four wider component bands and omit the all-zero BF16 reference row so the direction and magnitude of the spectral shift are easier to read. (b) End-to-end generation throughput for Qwen3-30B on 1× RTX PRO 6000 Blackwell. All runnable methods use full-precision KV cache, output length 1024, no prompt cache, no speculative decoding, and concurrency 4. Columns report ctx=1K/32K/128K; each cell shows tokens/s and speedup over the W8A8 baseline at the same context length. Sub-4-bit mixed precision is labeled N/A because it has no efficient accelerated GPU GEMM path.

Together with Table 4, Figure 2 (a) demonstrates that ours has the smallest signed deviations across most bands and the lowest L1/JSD distances to BF16, corresponding to the highest preservation score (0.9710). The lower $E@64$ and $E@128$ values show that the adapter redistributes a small amount of energy away from the leading components and better matches the full normalized spectrum.

5.3 System-Level Acceleration

We now evaluate whether the accuracy gains preserve the intended serving advantage. Beyond the predictable substantial GPU memory savings during training and inference brought by quantization

(e.g., quantizing a 30B MoE model by MXFP4 saves about 46.4 GB), Figure 2 (b) shows that the pure W4A8/MXFP4 tensor-core path improves decoding throughput by 1.38×, 1.53×, and 1.58× over W8A8 at 1K, 32K, and 128K context, respectively. LRR+LAC adds the low-rank adapter path needed for quality recovery, so it is slower than pure W4A8 by 6.1%, 7.0%, and 5.0% at the same context lengths. Even with this end-to-end adapter overhead, the deployed model remains 1.30×, 1.42×, and 1.50× faster than the W8A8 baseline.

A 32K-context kernel trace explains the gap. In W8A8 decoding, expert GEMMs account for 58% of per-token time, attention/KV for 22%, routing and dispatch for 12%, and other operations for 8%. MXFP4 reduces expert-GEMM time by 39%, while LRR+LAC adds 6.4% total time through adapter GEMMs and fusion overhead. The same trend holds at concurrency 1 and 8, where LRR+LAC is 1.31× and 1.45× faster than W8A8 at 32K context.

This throughput pattern also explains why we prefer rank adaptation over heterogeneous sub-4-bit formats. Because a low-rank adapter can be naturally fused into the group GEMMs pipeline alongside the expert that takes the same input, adjusting rank is more hardware accessible. Current GPU tensor-core expose efficient 4-bit and 8-bit matrix multiply, but not arbitrary 2-bit or 3-bit GEMM. In sparse MoE decoding, this gap is especially damaging: routed tokens are already fragmented by expert, and splitting expert batches again by bit width reduces tile occupancy, increases format-specific dispatches, complicates graph capture, and spends much of the nominal memory saving on packing, unpacking, and scale handling. LRR+LAC instead keeps every expert on the same W4A8 MXFP4 serving path and pays only the measured adapter overhead.

6 Conclusion

In summary, we presented a unified quantization framework for MoE models that jointly stabilizes routing and restores expert quality. By formulating quantization-induced route drift as rollout inconsistency under explicit route-margin conditions, we motivated limited routing replay for compression fine-tuning and provided conditional convergence and routing-consistency guarantees. We further introduced load-aware low-rank compensation with strict monotonic error-reduction guarantees under a fixed rank budget. Across modern MoE LLMs, the results show consistent improvements in perplexity, downstream accuracy, and route stability; in addition, the system study on Qwen3-30B shows that our W4A8 adapter runtime remains substantially faster than W8A8 and avoids unsupported mixed-precision sub-4-bit kernels.

Limitations. Limited replay creates a data-coverage trade-off. Caching more tokens gives better coverage of the fine-tuning distribution but increases preprocessing and storage cost. If the replay set is too small, adapter QAT can overfit replayed routes and may not fully transfer to unseen low-margin tokens.

Acknowledgments and Disclosure of Funding

This work was partially supported by Hong Kong CRF grants under Grant No. C7004-22G and C6015-23G.

References

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin

Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

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

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

OpenAI. gpt-oss-120b and gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Songkai Ma, Zhaorui Zhang, Sheng Di, Benben Liu, Xiaodong Yu, Xiaoyi Lu, and Dan Wang. Moe-compression: How the compression error of experts affects the inference accuracy of moe model?, 2025a. URL https://arxiv.org/abs/2509.07727.

Yuanteng Chen, Yuantian Shao, Peisong Wang, and Jian Cheng. EAC-MoE: Expert-selection aware compressor for mixture-of-experts large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12942–12963, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.633. URL https://aclanthology.org/2025.acl-long.633/.

Zhongqian Fu, Tianyi Zhao, Ning Ding, Xianzhi Yu, Xiaosong Li, Yehui Tang, and Yunhe Wang. Eaquant: Enhancing post-training quantization for moe models via expert-aware optimization, 2026. URL https://arxiv.org/abs/2506.13329.

Yi-Zeng Fang and Juinn-Dar Huang. Router choice matters: Rank-aware post-training quantization for moe models, 2025. URL https://openreview.net/forum?id=kPgLp47bJf.

Mohammed Nowaz Rabbani Chowdhury, Kaoutar El Maghraoui, Hsinyu Tsai, Naigang Wang, Geoffrey W. Burr, Liu Liu, and Meng Wang. Efficient quantization of mixture-of-experts with theoretical generalization guarantees. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yiMlVBAoQi.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

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 llm compression and acceleration, 2024. URL https://arxiv.org/abs/2306.00978.

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, 2025a. URL https://arxiv.org/abs/2405.16406.

Xiaohai Li, Xiaodong Yang, Yingwei Zhang, Jianrong Yang, and Yiqiang Chen. An adaptive joint optimization framework for pruning and quantization. *International Journal of Machine Learning and Cybernetics*, 15:5199–5215, 06 2024. doi: 10.1007/s13042-024-02229-w.

Pingzhi Li, Xiaolong Jin, Zhen Tan, Yu Cheng, and Tianlong Chen. Quantmoe-bench: Examining post-training quantization for mixture-of-experts, 2025. URL https://arxiv.org/abs/2406.08155.

Young Jin Kim, Raffy Fahim, and Hany Hassan Awadalla. Mixture of quantized experts (moqe): Complementary effect of low-bit quantization and robustness, 2023. URL https://arxiv.org/abs/2310.02410.

Longteng Zhang, Sen Wu, Shuai Hou, Zhengyu Qing, Zhuo Zheng, Danning Ke, Qihong Lin, Qiang Wang, Shaohuai Shi, and Xiaowen Chu. SALR: sparsity-aware low-rank representation for efficient fine-tuning of large language models. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor, editors, *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 28337–28345. AAAI Press, 2026. doi: 10.1609/AAAI.V40I33.40062. URL https://doi.org/10.1609/aaai.v40i33.40062.

Beichen Huang, Yueming Yuan, Zelei Shao, and Minjia Zhang. Milo: Efficient quantized moe inference with mixture of low-rank compensators, 2025. URL https://arxiv.org/abs/2504.02658.

Zhenyu Liu, Yunzhen Liu, Zehao Fan, Garrett Gagnon, Yayue Hou, Nan Wu, Yangwook Kang, and Liu Liu. Bandwidth-efficient adaptive mixture-of-experts via low-rank compensation, 2025b. URL https://arxiv.org/abs/2512.17073.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. Stabilizing moe reinforcement learning by aligning training and inference routers, 2025b. URL https://arxiv.org/abs/2510.11370.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Krishna Teja Chitty-Venkata, Jie Ye, and Murali Emani. Mopeq: Mixture of mixed precision quantized experts, 2025. URL https://arxiv.org/abs/2509.02512.

NVIDIA. Nvidia blackwell architecture technical brief, 2024. URL https://resources.nvidia.com/en-us-blackwell-architecture.

Zijie Yan and et al. Hongxiao Bai. Scalable training of mixture-of-experts models with megatron core, 2026. URL https://arxiv.org/abs/2603.07685.

Zhanhao Xie, Yuexiao Ma, Xiawu Zheng, Fei Chao, Wanchen Sui, Yong Li, Shen Li, and Rongrong Ji. Automated fine-grained mixture-of-experts quantization. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 27024–27037, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1386. URL https://aclanthology.org/2025.findings-acl.1386/.

Sebastián Andrés Cajas Ordóñez, Luis Fernando Torres Torres, Mackenzie J. Meni, Carlos Andrés Duran Paredes, Eric Arazo, Cristian Bosch, Ricardo Simon Carbajo, Yuan Lai, and Leo Anthony Celi. Uncertainty makes it stable: Curiosity-driven quantized mixture-of-experts, 2026. URL https://arxiv.org/abs/2511.11743.

NVIDIA, 2025. URL https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation, 2024a. URL https://arxiv.org/abs/2407.05000.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized?, 2024b. URL https://arxiv.org/abs/2407.18242.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL https://arxiv.org/abs/1811.00937.

Appendix

A Proof of Theoretical Results

A.1 Proof of Proposition 3.1

Proposition. *Suppose both settings share identical token distribution $\mathcal{D}$ and top-k selection rule, and fix a logit gauge by subtracting the mean logit from each vector. Assume the centered router logits are bounded and that the BF16 route margin*

$$\gamma(x) = s^{\inf}_{(k)}(x) - s^{\inf}_{(k+1)}(x) \tag{29}$$

is positive on the tokens of interest. If

$$\|\boldsymbol{s}^{\inf}(x) - \boldsymbol{s}^{\mathrm{tr}}(x)\|_\infty < \gamma(x)/2, \tag{30}$$

then the top-k route is unchanged. Consequently, any objective that controls centered logit drift, or controls policy drift together with an explicit inverse-stability condition on the softmax map, upper-bounds quantization route mismatch on large-margin tokens. In this sense, policy inconsistency is a useful surrogate for route inconsistency, but not an unconstrained optimization-order equivalent objective.

Proof. We first consider a fixed x. If every logit changes by less than $\gamma(x)/2$ in ℓ_∞ norm, then the k-th BF16 logit remains larger than the $(k+1)$-st BF16 logit after perturbation, so no expert can cross the top-k boundary. Therefore route mismatch can occur only when the centered logit perturbation reaches at least half the BF16 margin:

$$\mathbf{1}\{\mathcal{I}_{\inf}(x) \neq \mathcal{I}_{\mathrm{tr}}(x)\} \leq \mathbf{1}\{\|\boldsymbol{s}^{\inf}(x) - \boldsymbol{s}^{\mathrm{tr}}(x)\|_\infty \geq \gamma(x)/2\}. \tag{31}$$

Taking expectation gives a margin-dependent upper bound on Δ_{route}. To connect this route event with policy drift, Pinsker's inequality gives

$$\|\pi_{\inf} - \pi_{\mathrm{tr}}\|_1^2 \leq 2\mathrm{KL}(\pi_{\inf}\|\pi_{\mathrm{tr}}), \tag{32}$$

so KL controls total-variation policy drift. However, softmax Lipschitzness alone only maps logit drift to policy drift and does not imply the inverse direction. The final policy-to-route implication therefore additionally requires an inverse-stability condition on the bounded, gauge-fixed logit region under consideration. Under that condition, controlling policy inconsistency controls the centered logit perturbation and hence route mismatch on large-margin tokens. □

A.2 Proof of Theorem 4.1

Theorem. *Under the assumption above, gradient descent*

$$\phi^{t+1} = \phi^t - \eta\nabla F(\phi^t), \qquad 0 < \eta \leq \frac{1}{L}, \tag{33}$$

converges linearly:

$$F(\phi^t) - F^\star \leq (1-\eta\mu)^t\big(F(\phi^0) - F^\star\big). \tag{34}$$

Proof. We start from the standard descent inequality. By L-smoothness,

$$F(\phi^{t+1}) \leq F(\phi^t) - \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla F(\phi^t)\|_2^2. \tag{35}$$

With $\eta \leq 1/L$,

$$F(\phi^{t+1}) \leq F(\phi^t) - \frac{\eta}{2}\|\nabla F(\phi^t)\|_2^2 \leq F(\phi^t) - \eta\mu\big(F(\phi^t) - F^\star\big), \tag{36}$$

where the second inequality uses PL. Rearranging yields

$$F(\phi^{t+1}) - F^\star \leq (1-\eta\mu)(F(\phi^t) - F^\star), \tag{37}$$

and recursion gives the claim. □

A.3 Proof of Theorem 4.2

Theorem. *On replayed tokens $\{x_t\}$, limited replay enforces exact routing consistency:*

$$\mathcal{I}_{\text{train}}(x_t) = \mathcal{I}_t^\star, \quad \boldsymbol{\pi}_{\text{train}}(x_t) = \boldsymbol{\pi}_t^\star \quad \forall t. \tag{38}$$

If in deployment router logits satisfy

$$\|\widehat{\boldsymbol{s}}(x) - \boldsymbol{s}^\star(x)\|_\infty \le \varepsilon \tag{39}$$

with probability at least $1-\delta$, and route margin $\gamma(x) \ge 2\varepsilon$ with probability at least $1-\rho$, then

$$\Pr\left[\widehat{\mathcal{I}}(x) = \mathcal{I}^\star(x)\right] \ge 1 - (\delta + \rho). \tag{40}$$

Proof. The first part is by construction of replay. For the second part, whenever both events hold: (i) logit perturbation is bounded by ε, and (ii) the margin is at least 2ε, no pair can cross the top-k boundary, so $\widehat{\mathcal{I}}(x) = \mathcal{I}^\star(x)$. By the union bound, failure probability is at most $\delta + \rho$. □

A.4 Proof of Theorem 4.3

Theorem. *Assume for every allocation step $t < R$, there exists some expert j with $\sigma_{j,r_j^{(t)}+1} > 0$. Let $\mathbf{r}^{(t)}$ be greedy load-aware allocation after t rank units. Then*

$$\mathcal{E}(\mathbf{r}^{(t+1)}) < \mathcal{E}(\mathbf{r}^{(t)}) \quad \forall t < R. \tag{41}$$

Hence weighted compression error is strictly decreasing with budget.

Proof. At step t, greedy picks $j_t \in \arg\max_j \Delta_j(r_j^{(t)})$. By assumption, maximum gain is positive: $\Delta_{j_t}(r_{j_t}^{(t)}) > 0$. Since only r_{j_t} increases by one, we have

$$\mathcal{E}(\mathbf{r}^{(t+1)}) = \mathcal{E}(\mathbf{r}^{(t)}) - \Delta_{j_t}(r_{j_t}^{(t)}), \tag{42}$$

which is strictly smaller. □

A.5 Proof of Proposition 4.4

Proposition. *For any fixed budget R, an allocation $\mathbf{r}^\star$ is globally optimal iff the selected set of R unit gains*

$$\{p_j \sigma_{j,m}^2 : 1 \le m \le r_j^\star\} \tag{43}$$

contains the R largest values among all unit gains $\{p_j\sigma_{j,m}^2\}_{j,m}$.

Proof. To make the selection structure explicit, rewrite the objective as

$$\mathcal{E}(\mathbf{r}) = \sum_j p_j \sum_{m \ge 1} \sigma_{j,m}^2 - \sum_j \sum_{m=1}^{r_j} p_j \sigma_{j,m}^2. \tag{44}$$

The first term is constant, so minimizing error is equivalent to maximizing the sum of selected unit gains under cardinality R, which is solved by taking the largest R gains. □

B Route-Consistent Adaptation

Putting the two components together, we summarize the full pipeline that couples limited routing replay and load-aware compensation in Algorithm 1.

Algorithm 1 Routing Replay with Load-Aware Compensation (LRR+LAC)

Require: BF16 MoE model $\mathcal{M}_{\text{bf16}}$; replay/QAT set $\mathcal{D}_{\text{rep}} = \{(x_t, y_t)\}_{t=1}^{N}$; top-$k$; experts M; average rank $\bar{r} = 128$ (thus $R = 128M$); maximal per expert rank $\hat{r} = 256$, optimization steps T; learning rate η

Ensure: Compressed model $\mathcal{M}_{\text{mxfp4}}$ with replay-aligned routing and adapter-only QAT

One-shot replay cache/workload pass:
for each token-label pair $(x_t, y_t) \in \mathcal{D}_{\text{rep}}$ **do**
    Run BF16 router, obtain $(\mathcal{I}_t^\star, \boldsymbol{\pi}_t^\star)$, append $(x_t, y_t, \mathcal{I}_t^\star, \boldsymbol{\pi}_t^\star)$ to cache $\mathcal{C}$
    Update workload counters for all experts in $\mathcal{I}_t^\star$
end for
Compute normalized loads $p_j \leftarrow \frac{1}{Nk} \sum_{t=1}^{N} \mathbf{1}\{j \in \mathcal{I}_t^\star\}$
Pre-QAT rank allocation:
    For residuals $E_j = W_j - Q(W_j)$ and singular values $\{\sigma_{j,m}\}$, greedily allocate ranks under $\sum_j r_j \le R$ and $r_j \le \hat{r}$ by maximizing $\Delta_j(r_j) = p_j \sigma_{j,r_j+1}^2$
MXFP4 initialization: quantize experts $\widehat{W}_j \leftarrow Q(W_j)$; instantiate rank-$r_j$ low-rank adapters; freeze router and all base MXFP4 weights
for $t = 1$ to T **do**
    Sample minibatch $\{(x_b, y_b, \mathcal{I}_b^\star, \boldsymbol{\pi}_b^\star)\}_{b=1}^{B}$ from cache $\mathcal{C}$
    Replay the cached BF16 routes $(\mathcal{I}_b^\star, \boldsymbol{\pi}_b^\star)$ for each token
    Compute replayed loss $F(\phi)$
    Update adapters only: $\phi \leftarrow \phi - \eta \nabla_\phi F(\phi)$
end for
Export final $\mathcal{M}_{\text{mxfp4}}$

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: See Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors

should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: See theorems.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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: See Section B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [No]

   Justification: no justifications.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: See Section 5.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: [No]

   Justification: error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: See Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: there is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: the paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: See Section 5.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: the paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: that the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.