PROFIMOE: PROMOTION-AWARE MULTI-FIDELITY INFERENCE FOR MIXTURE-OF-EXPERTS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-routing compression of mixture-of-experts (MoE) models often saves computation by skipping selected routed experts, removing their token-specific responses. We introduce full-or-light execution: each routed expert runs either its Full Expert or a lower-cost Light Subnet. Since Light execution already preserves part of the expert response, the relevant allocation signal is the magnitude of the response recovered by upgrading an expert from Light to Full. We call this signal Promotion Utility. ProFiMoE applies this principle to frozen pretrained MoE models without retraining. It constructs contiguous Light Subnets from calibration-ranked intermediate channels and estimates Promotion Utility by combining an expert-contribution prior with a calibrated Fidelity Gap. Promotion-Aware Allocation then assigns Full execution through a cumulative Top-P rule, while Observed-Mass Rescaling adjusts the mixed-fidelity output using calibrated Prefix Coverage. We evaluate three MoE models at target effective FFN pruning ratios from 10% to 60%. At 50% and 60% target pruning on Qwen3, the reported seven-task mean is 2.83 and 2.81 points above the ACE reference. For 1024-token inputs on an NVIDIA A100 at 60% target pruning, time to first token and time per output token decrease by up to 50.7% and 22.7%, respectively, relative to BF16.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) models control per-token computation by routing each input to only a small subset of experts, enabling large parameter capacity without activating the full model for every token (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). However, this sparsity largely stops at expert selection: once an expert is routed, its full feed-forward network is typically evaluated, leaving substantial intra-expert computation untouched. Existing post-routing compression methods further reduce computation by selectively skipping routed experts, but many existing approaches still formulate compression as a binary execution decision: each routed expert either executes in full or is removed from the current token computation (Lu et al., 2024; Zhao et al., 2025; Lasby et al., 2025; Liu et al., 2026b). We refer to this conventional execution space as *full-or-zero*.

The full-or-zero abstraction leaves an intermediate execution regime unexplored: a routed expert may not require full computation, yet discarding its response entirely can discard useful token-specific information. Instead of deciding only whether a routed expert executes, we consider how faithfully it should execute. For experts that are not assigned Full computation, we retain a compact *Light Subnet* formed from a subset of informative intermediate channels, reusing the pretrained weights and current token activations. This yields a *full-or-light* execution space in which every routed expert preserves a token-specific response, while only selected experts incur the cost of full-width execution.

This richer execution space changes the allocation objective. Under full-or-zero inference, an expert's full-response contribution is a natural importance signal because selection determines whether its routed response is present. Under full-or-light inference, the Light Subnet already contributes a partial response, so the relevant quantity is the additional response recovered by upgrading it from Light to Full. We call this marginal quantity *Promotion Utility*. A large Full response may need little promotion when the Light Subnet is faithful, whereas a smaller response may deserve promotion when Light execution misses more of it. Thus, Promotion Utility provides a more suitable allocation signal than Full Contribution alone in the full-or-light setting.

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

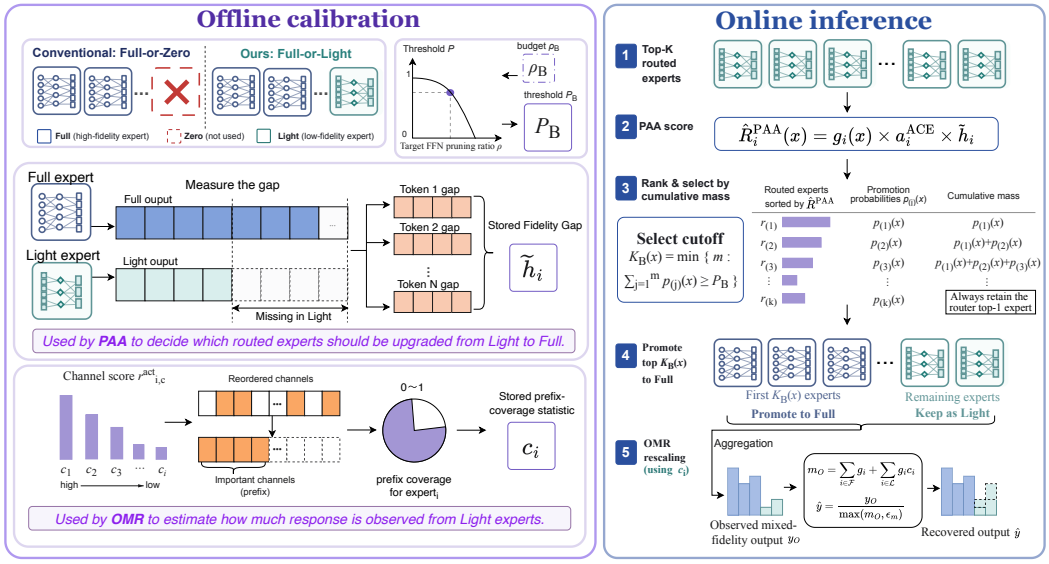


Figure 1: Overview of ProFiMoE. Offline calibration constructs Light Subnets and estimates Fidelity Gap and Prefix Coverage. Online, PAA allocates Full/Light execution via Promotion Utility and cumulative Top-P, while OMR rescales the mixed-fidelity output using estimated Observed Mass.

We instantiate this principle in ProFiMoE, a post-routing multi-fidelity framework for frozen pretrained MoE models. ProFiMoE leaves the pretrained router unchanged and introduces a second-stage fidelity allocation over the routed experts. Offline, it reorders each expert's informative intermediate channels into a contiguous prefix, yielding a structure-aligned Light Subnet while preserving the original Full Expert exactly. It also calibrates two complementary statistics: the *Fidelity Gap*, which characterizes the expected Light-to-Full discrepancy, and *Prefix Coverage*, which estimates how much response mass remains observable under Light execution. Online, *Promotion-Aware Allocation* (PAA) combines a routed full-contribution prior with the calibrated Fidelity Gap to estimate each expert's Promotion Utility. A cumulative Top-P rule ranks experts by Promotion Utility and promotes the smallest prefix that reaches a score threshold calibrated to the target FFN budget.

The full-or-light execution space also changes the form of the aggregated expert response. Unlike full-or-zero execution, where skipped experts contribute nothing, Light Experts retain a partial response that changes the scale of the mixed-fidelity aggregate. We therefore introduce *Observed-Mass Rescaling* (OMR), which uses calibration-derived Prefix Coverage to estimate the fraction of routed response mass preserved under the current Full/Light assignment and rescales the aggregate according to the estimated Observed Mass. PAA and OMR address complementary aspects of multi-fidelity inference: PAA determines where Full computation is most valuable, while OMR calibrates the response produced by the resulting fidelity mixture.

Figure 1 summarizes the resulting offline-to-online pipeline. Together, these components turn post-routing MoE compression from a binary expert-retention problem into a token-adaptive fidelity-allocation problem. Our main contributions are as follows.

- We formulate post-routing MoE compression as a full-or-light fidelity-allocation problem and identify Promotion Utility—the additional response recovered by upgrading a routed expert from Light to Full—as a marginal allocation objective for full-or-light inference.
- We introduce ProFiMoE, a training-free multi-fidelity inference framework for frozen pretrained MoE models. ProFiMoE preserves the pretrained router and Full Expert functions, while combining structure-aligned Light Subnets, *Promotion-Aware Allocation*, and *Observed-Mass Rescaling* to allocate and calibrate mixed-fidelity expert execution.
- We bound the mixed-fidelity residual by the Promotion Utilities of Light-executed experts and analyze cumulative allocation and rescaling error. At 50% and 60% target pruning on Qwen3, the reported seven-task mean is 2.83 and 2.81 points above the ACE reference.

2 Related Work

Efficient MoE inference and expert selection. Sparse MoE models limit per-token computation by routing each input to a small expert subset (Shazeer et al., 2017). GShard, Switch Transformer, and ST-MoE establish this paradigm through distributed sharding, simplified routing, and stable sparse training (Lepikhin et al., 2021; Fedus et al., 2022; Zoph et al., 2022). Subsequent systems adapt routed computation to token difficulty, router confidence, layer sensitivity, or a global budget (Huang et al., 2024a; Zhao et al., 2025; Guo et al., 2025; Liu et al., 2026a). These methods primarily decide which experts enter the active set. Post-routing approaches instead retain the pretrained router and reduce the active set using routing frequency, activation statistics, or estimated contribution (Lu et al., 2024; Xie et al., 2024; Lasby et al., 2025; Liu et al., 2026b). Their execution decision is typically full-or-zero. ProFiMoE expands the action space after routing: every selected expert remains active, but each executes through either its Full Expert or its Light Subnet. ACE (Xu et al., 2026) supplies the structural full-contribution prior used in our experiments; PAA combines this prior with upgrade necessity and ranks experts by the marginal value of promotion.

Intra-expert compression and multi-fidelity execution. Intra-expert compression reduces the cost of each expert rather than the number of experts executed. One-shot pruning reconstructs compressed responses or compensates for removed channels while retaining matrix-friendly structure (Frantar & Alistarh, 2023; An et al., 2024); MoE-specific methods allocate nonuniform expert capacity, exploit layer-wise redundancy, or combine expert-aware quantization with online reduction (Bai et al., 2025; Huang et al., 2024b). These approaches generally install one compressed representation at a fixed fidelity. Conditional-computation models instead expose several computation levels: CoLT5 assigns expensive feed-forward and attention paths to selected tokens (Ainslie et al., 2023), and Mixture of Nested Experts learns nested expert sizes along a compute–accuracy frontier (Jain et al., 2024). ProFiMoE brings multi-fidelity execution to frozen, pretrained MoE experts. A function-preserving channel permutation exposes a contiguous Light Subnet inside every Full Expert, allowing fidelity to be assigned independently for each routed expert and token.

Information preservation and output rescaling. Compression changes both the magnitude and direction of the model response. Reconstruction-based pruning and bias compensation adapt the retained computation after parameters or channels are removed (Frantar & Alistarh, 2023; An et al., 2024), while token reorganization preserves low-priority visual information by fusing representations instead of deleting them (Liang et al., 2022). In expert skipping, gate renormalization rescales the surviving expert outputs; the skipped experts contribute no token-specific observation. ProFiMoE retains a partial response from every routed expert through Light execution. This makes output correction an observed-mass problem: PAA decides which partial responses should be upgraded, and OMR rescales the resulting Full/Light aggregate using calibrated Prefix Coverage. The separation of promotion and rescaling is specific to a mixed-fidelity execution space and distinguishes ProFiMoE from both expert removal and static compressed-expert replacement.

3 Method

3.1 Overview

ProFiMoE is a post-routing multi-fidelity framework for frozen pretrained MoE models. It leaves the pretrained router and Full Expert functions unchanged, and converts a routed MoE layer from binary expert execution into token-adaptive Full/Light execution. Consider a layer with N experts indexed by $i \in \{1, \dots, N\}$. For token state $x \in \mathbb{R}^d$, where d is the model hidden dimension, the pretrained router selects a set $\mathcal{R}(x)$ of k experts and produces nonnegative normalized gates $g_i(x)$ with $\sum_{i \in \mathcal{R}(x)} g_i(x) = 1$. Let $f_i(x) \in \mathbb{R}^d$ be the Full Expert output and $y^*(x)$ the corresponding routed output:

$$y^*(x) = \sum_{i \in \mathcal{R}(x)} g_i(x) f_i(x). \tag{1}$$

For each routed expert, a fidelity variable $M_i(x) \in \{0, 1\}$ chooses between the Full Expert ($M_i = 1$) and a width-S Light Subnet ($M_i = 0$); when x is fixed, we write M_i for $M_i(x)$. Here, d_{ff} denotes

the intermediate width of the Full Expert, $S < d_{\text{ff}}$ is the Light Subnet width, and $f_i^S(x)$ denotes its output. Collecting the assignments into the mask $M = (M_i(x))_{i\in\mathcal{R}(x)}$, we denote the resulting mixed-fidelity observation by $y_O(x, M)$:

$$y_O(x, M) = \sum_{i\in\mathcal{R}(x)} g_i(x)\big[M_i f_i(x) + (1 - M_i) f_i^S(x)\big]. \tag{2}$$

We denote its residual relative to Full execution by $r(x, M)$:

$$r(x, M) = y^*(x) - y_O(x, M) = \sum_{i\in\mathcal{R}(x):M_i=0} g_i(x)\big[f_i(x) - f_i^S(x)\big]. \tag{3}$$

The residual in Eq. (3) exposes the quantity that allocation should preserve. Under full-or-zero execution, adding expert i recovers $g_i(x)f_i(x)$. We denote the magnitude of this Full Contribution by $C_i(x)$:

$$C_i(x) = g_i(x)\|f_i(x)\|_2. \tag{4}$$

Under full-or-light execution, promotion adds only $g_i(x)\big(f_i(x) - f_i^S(x)\big)$. We define its magnitude as the Promotion Utility $R_i(x)$:

$$R_i(x) = g_i(x)\|f_i(x) - f_i^S(x)\|_2. \tag{5}$$

With $\epsilon > 0$ as a numerical stabilizer, we denote the normalized Light-to-Full discrepancy by the Fidelity Gap $\eta_i(x)$:

$$\eta_i(x) = \frac{\|f_i(x) - f_i^S(x)\|_2}{\|f_i(x)\|_2 + \epsilon}. \tag{6}$$

Up to the stabilizer ϵ, $R_i(x) = C_i(x)\eta_i(x)$. This factorization shows that the benefit of promoting expert i depends on both the magnitude of its Full response and the fraction of that response missed by Light execution. A routed expert may have a large Full Contribution but require little additional computation when its Light approximation is faithful; conversely, an expert with a more modest Full Contribution may deserve promotion when Light execution misses a large fraction of its response. Thus, Promotion Utility ranks Light-to-Full upgrades by their marginal benefit, whereas Full Contribution measures only the magnitude of the Full response. Since evaluating $R_i(x)$ exactly would already require both fidelities, PAA estimates its two factors before expert execution. Figure 1 summarizes the pipeline: offline calibration builds Light Subnets, estimates Fidelity Gaps and Prefix Coverage, and maps channel-equivalent FFN budgets to promotion thresholds. Online, PAA assigns Full/Light fidelity from Promotion Utility, and OMR rescales the aggregate by Observed Mass.

3.2 Offline Calibration

Offline Calibration prepares the structures and statistics required for online fidelity allocation. It serves three purposes: constructing a contiguous Light Subnet for each expert, estimating the expert-level statistics used by PAA and OMR, and calibrating the mapping from a target channel-equivalent FFN budget to the promotion threshold used online.

3.2.1 Structure-Aligned Light Subnet Construction

For a SwiGLU expert, let $W_{g,i}, W_{u,i} \in \mathbb{R}^{d_{\text{ff}}\times d}$ be the gate and up-projection matrices, and let $W_{d,i} \in \mathbb{R}^{d\times d_{\text{ff}}}$ be the down-projection matrix. Its Full Expert output is

$$f_i(x) = W_{d,i}\big[\,\text{SiLU}(W_{g,i}x) \odot (W_{u,i}x)\big]. \tag{7}$$

Let $\mathcal{D}_{\text{cal}}$ be the calibration set, and let $c \in \{1, \ldots, d_{\text{ff}}\}$ index an intermediate channel. We define the routed channel score $r_{i,c}^{\text{act}}$ from its typical activation and its transmission through the down projection:

$$r_{i,c}^{\text{act}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}|i\in\mathcal{R}(x)}\left[|\text{SiLU}(W_{g,i}x)_c(W_{u,i}x)_c|\right]\|W_{d,i}[:, c]\|_2. \tag{8}$$

Let $z_{i,c}(x) = \text{SiLU}(W_{g,i}x)_c(W_{u,i}x)_c$ denote the post-SwiGLU activation of channel c. The expectation term $\mathbb{E}[|z_{i,c}(x)|]$ measures its average activation magnitude on tokens routed to expert i, while $\|W_{d,i}[:, c]\|_2$ measures how strongly one unit of that activation is projected into the output space. Their product equals the expected norm of channel c's individual output contribution,

$\mathbb{E}[\|W_{d,i}[:,c]z_{i,c}(x)\|_2]$. Thus, $r_{i,c}^{\text{act}}$ ranks channels by their expected individual contribution to the expert output on tokens routed to expert i, which we use to construct the Light Subnet.

Let π_i sort channels in descending $r_{i,c}^{\text{act}}$. We apply the same permutation to both input projections and the matching column permutation to the down projection:

$$W_{g,i} \leftarrow W_{g,i}[\pi_i,:], \quad W_{u,i} \leftarrow W_{u,i}[\pi_i,:], \quad W_{d,i} \leftarrow W_{d,i}[:,\pi_i]. \tag{9}$$

Because SwiGLU acts independently along the intermediate-channel dimension, applying the same permutation to the gate and up projections and the matching column permutation to the down projection is function preserving: the Full Expert output remains exactly unchanged. After reordering, we relabel the channels and their scores by the new positions, so the first S channels form the contiguous prefix $\mathcal{C}_i^S = \{1,\dots,S\}$, and define

$$f_i^S(x) = W_{d,i}[:,\mathcal{C}_i^S]\left[\text{SiLU}(W_{g,i}[\mathcal{C}_i^S,:]x) \odot W_{u,i}[\mathcal{C}_i^S,:]x\right]. \tag{10}$$

Thus each Light Subnet is embedded directly in its Full Expert, reuses the original parameters and token activations, and executes as a contiguous width-S dense GEMM. The construction introduces no retraining and preserves the Full Expert function exactly.

3.2.2 Calibrating the Fidelity Gap and Prefix Coverage

PAA needs upgrade information before running both fidelities. We denote the mean Fidelity Gap over calibration tokens routed to expert i by h_i^{FG}:

$$h_i^{\text{FG}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{cal}}|i\in\mathcal{R}(x)}[\eta_i(x)]. \tag{11}$$

The reliability of an expert-level Fidelity Gap estimate depends on its calibration support. We shrink each estimate toward the corresponding layer mean according to the number of routed calibration tokens. Let N_i be the number of calibration tokens routed to expert i, let $\ell(i)$ denote its layer, and let $\bar{h}_{\ell(i)}$ be the mean Fidelity Gap in that layer. Given a shrinkage scale $\tau > 0$, let λ_i denote the expert-specific shrinkage weight and $\widetilde{h}_i$ the stored Fidelity Gap estimate:

$$\widetilde{h}_i = \lambda_i h_i^{\text{FG}} + (1-\lambda_i)\bar{h}_{\ell(i)}, \qquad \lambda_i = \frac{N_i}{N_i+\tau}, \tag{12}$$

Experts with abundant routed calibration tokens retain their expert-specific estimate, whereas sparsely observed experts rely more strongly on the layer-level statistic. The stored $\widetilde{h}_i$ is the expert-level Fidelity Gap used by PAA to modulate promotion.

OMR uses a distinct statistic. We denote the fraction of offline channel-importance mass retained by the Light Subnet as its Prefix Coverage c_i:

$$c_i = \frac{\sum_{c\in\mathcal{C}_i^S} r_{i,c}^{\text{act}}}{\sum_{c=1}^{d_{\text{ff}}} r_{i,c}^{\text{act}} + \epsilon}. \tag{13}$$

Prefix Coverage c_i measures the fraction of offline channel-importance mass retained by the Light Subnet. We use it as a calibration-derived proxy for the fraction of the Full response that remains observable under Light execution. The two statistics serve distinct roles. The stored Fidelity Gap $\widetilde{h}_i$ is estimated from paired Full/Light outputs and characterizes how much of expert i's response is typically missed by Light execution; PAA therefore uses it to estimate the value of promotion. Prefix Coverage c_i, in contrast, serves as the observable-fraction proxy used by OMR. Thus, Fidelity Gap governs allocation, whereas Prefix Coverage governs output calibration.

3.2.3 Budget-to-Coverage Calibration

The cumulative Top-P threshold controls promotion-score coverage rather than computation directly. We therefore calibrate the threshold P_B so that its expected Full/Light assignment matches a requested channel-equivalent FFN budget. For a target budget setting B, let ρ_B be the requested effective FFN pruning ratio and P_B the promotion-coverage threshold used online. Each token has $k = |\mathcal{R}(x)|$ routed experts. Because every Light Subnet still evaluates S channels, the resulting cost depends on

how many of these k experts are promoted. We denote the channel-equivalent FFN cost of mask M by $C(M)$ and the incremental cost of one promotion by $\Delta C = d_{\text{ff}} - S$:

$$C(M) = \sum_{i \in \mathcal{R}(x)} \big[M_i d_{\text{ff}} + (1 - M_i) S \big], \qquad \Delta C = d_{\text{ff}} - S, \tag{14}$$

The corresponding effective FFN pruning ratio $\rho_{\text{FFN}}(M)$ is

$$\rho_{\text{FFN}}(M) = 1 - \frac{C(M)}{k d_{\text{ff}}}. \tag{15}$$

For a candidate threshold $P \in (0, 1]$, let $M_P(x)$ denote the final Full/Light mask produced at threshold P, including the router top-1 inclusion defined below. For each target ρ_B, we search over P on the calibration workload and choose P_B such that

$$\mathbb{E}_{x \sim \mathcal{D}_{\text{cal}}} \big[\rho_{\text{FFN}}\big(M_{P_B}(x)\big) \big] \approx \rho_B. \tag{16}$$

We fix the calibrated map $\rho_B \mapsto P_B$ for evaluation. The resulting fixed map converts a requested channel-equivalent FFN reduction into the promotion-score coverage used by PAA at inference time.

3.3 Online Inference

3.3.1 Promotion-Aware Allocation (PAA)

For token x, the pretrained router returns $\{(i, g_i(x))\}_{i \in \mathcal{R}(x)}$. Let a_i^{ACE} denote the stored expert-level contribution prior from ACE (Xu et al., 2026). We combine this prior with the current router gate to estimate Full Contribution:

$$\widehat{C}_i(x) = g_i(x) a_i^{\text{ACE}}. \tag{17}$$

Combining this token-conditioned contribution proxy with the calibrated Fidelity Gap $\widetilde{h}_i$ gives the estimated Promotion Utility

$$\widehat{R}_i^{\text{PAA}}(x) = \widehat{C}_i(x) \widetilde{h}_i = g_i(x) a_i^{\text{ACE}} \widetilde{h}_i. \tag{18}$$

The resulting score is high when the router assigns substantial mass to an expert, its stored contribution prior is large, and Light execution is expected to incur a large Fidelity Gap. PAA uses $\widehat{R}_i^{\text{PAA}}(x)$ as a pre-execution proxy for the response recovered by promotion.

We normalize the estimated Promotion Utilities over the routed set to obtain the promotion score $p_i(x)$:

$$p_i(x) = \frac{\widehat{R}_i^{\text{PAA}}(x)}{\sum_{j \in \mathcal{R}(x)} \widehat{R}_j^{\text{PAA}}(x) + \epsilon}, \tag{19}$$

After sorting $p_i(x)$ in descending order as $p_{(1)}(x) \geq \cdots \geq p_{(k)}(x)$, PAA selects the shortest prefix whose cumulative score reaches P_B:

$$K_B(x) = \min \left\{ m \in \{1, \ldots, k\} : \sum_{j=1}^{m} p_{(j)}(x) \geq P_B \right\}. \tag{20}$$

The first $K_B(x)$ experts form $\mathcal{F}(x)$ and execute in Full; the remainder form $\mathcal{L}(x)$ and execute in Light. Concentrated promotion mass reaches P_B with fewer Full Experts, while diffuse mass allocates Full computation more broadly. We also retain the router's top-1 expert in $\mathcal{F}(x)$. Because every promotion has the same incremental cost $d_{\text{ff}} - S$, the cumulative Top-P prefix reaches P_B with the fewest promotions and the lowest incremental channel-equivalent cost. The router top-1 inclusion can add one Full Expert beyond this prefix. The minimum-cardinality result in Appendix A therefore applies to the underlying cumulative allocation rule before the top-1 inclusion. The resulting assignment defines the final mask $M_{P_B}(x)$.

3.3.2 MIXED-FIDELITY EXECUTION AND OBSERVED-MASS RESCALING (OMR)

After PAA assigns fidelity, Full and Light kernels evaluate their respective expert sets. With threshold P_B fixed, we write $y_O(x) = y_O(x, M_{P_B}(x))$ for their aggregated token-specific response:

$$y_O(x) = \sum_{i \in \mathcal{F}(x)} g_i(x) f_i(x) + \sum_{i \in \mathcal{L}(x)} g_i(x) f_i^S(x). \tag{21}$$

To model the leading scale distortion introduced by Light execution, we use a first-order proportional observation model. For each Light Expert, we write

$$f_i^S(x) = c_i f_i(x) + \varepsilon_i(x), \qquad c_i \in [0, 1], \tag{22}$$

where c_i is the calibration-derived Prefix Coverage and $\varepsilon_i(x) \in \mathbb{R}^d$ absorbs the residual directional and proportional mismatch not explained by this scalar approximation. Under this model, we assign unit observation coefficient to Full Experts and the calibrated coefficient c_i to Light Experts. This yields the estimated gate-weighted Observed Mass

$$m_O(x) = \sum_{i \in \mathcal{F}(x)} g_i(x) + \sum_{i \in \mathcal{L}(x)} g_i(x) c_i. \tag{23}$$

Let $\epsilon_m > 0$ be a numerical floor for the rescaling denominator and let $\widehat{y}(x)$ denote the recovered output. OMR computes

$$\widehat{y}(x) = \frac{y_O(x)}{\max(m_O(x), \epsilon_m)}. \tag{24}$$

Under the proportional observation model, OMR compensates for the estimated scale attenuation while reducing the dominant magnitude distortion introduced by Light execution.Appendix A characterizes the remaining error as residual proportional mismatch divided by estimated Observed Mass. Full and Light execution retain regular dense GEMM shapes. Outside the expert kernels, the online controller performs expert-level scalar scoring, sorting and cumulative selection over the routed top-k set, accumulation of the estimated Observed Mass, and one hidden-vector rescaling per MoE layer.

4 EXPERIMENTS

4.1 SETUP

We evaluate Qwen3-30B-A3B-Instruct-2507, Qwen3.6-35B-A3B, and Gemma-4-26B-A4B-it, covering different routing designs, expert granularities, active parameter counts, and model families (Yang et al., 2025; Qwen Team, 2026; Gemma Team, 2026). For language modeling, we report WikiText-2 perplexity at sequence length 2048 on Qwen3 and Qwen3.6 (Merity et al., 2017). For downstream evaluation, we use seven tasks spanning multiple capability domains: PIQA for physical commonsense (Bisk et al., 2020), ARC-Easy and ARC-Challenge for science reasoning (Clark et al., 2018), MATH-500 for mathematical reasoning (Hendrycks et al., 2021), GPQA-Diamond for graduate-level science reasoning (Rein et al., 2024), and HumanEval and LiveCodeBench for code generation (Chen et al., 2021; Jain et al., 2025). We report an unweighted average over these seven downstream tasks, with ARC-Easy and ARC-Challenge counted separately.

We evaluate target effective FFN pruning ratios from 10% to 60% in increments of 10%. For each target ratio ρ_B, the promotion threshold P_B is calibrated once on the calibration workload using Eq. (16) and then fixed throughout evaluation. All reported runs use BF16 weights. Baseline results follow the ACE evaluation record, while ProFiMoE results come from the present runs. Table 1 compares the reported means at common target FFN pruning ratios.

We compare ProFiMoE with router-score ranking, NAEE (Lu et al., 2024), MoDES (Huang et al., 2026), DiEP (Bai et al., 2025), and ACE (Xu et al., 2026). Table 1 reports results across all three models and six target pruning ratios; the appendix gives the task-level ProFiMoE scores. ACE skips selected routed experts, whereas ProFiMoE retains Light execution for non-promoted experts. Component-level analysis is conducted on Qwen3.5-122B at 10% and 15% effective FFN pruning. WikiText-2 perplexity is reported separately, while the component-analysis mean is computed over six downstream accuracy tasks.

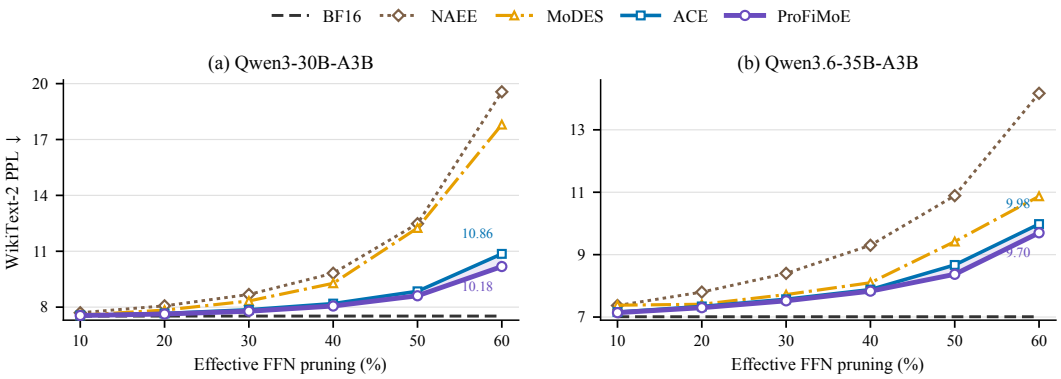


Figure 2: WikiText-2 perplexity across effective FFN pruning ratios on Qwen3-30B-A3B and Qwen3.6-35B-A3B. BF16 denotes the unpruned model; panels use model-specific y-axis ranges.

4.2 Main results

Table 1: Average downstream accuracy (%) over seven tasks at different effective FFN pruning ratios. ARC-Easy and ARC-Challenge are counted separately. Score denotes router-score ranking. Dense is excluded from compressed-method ranking; bold and underline denote the best and second-best compressed methods, respectively. Δ is the difference between the reported ProFiMoE and ACE means at the same target pruning ratio.

Model	Ratio	Dense	Score	NAEE	MoDES	DiEP	ACE	ProFiMoE	Δ vs. ACE
Qwen3-30B-A3B	10%	80.67	79.69	80.39	79.79	76.58	80.39	**80.77**	+0.38
	20%	80.67	77.85	79.34	78.80	64.36	80.25	**80.50**	+0.25
	30%	80.67	77.97	77.77	77.99	47.05	79.77	**79.79**	+0.02
	40%	80.67	75.39	72.93	76.03	27.44	78.21	**78.50**	+0.29
	50%	80.67	66.69	64.40	56.61	3.96	74.30	**77.13**	**+2.83**
	60%	80.67	37.64	43.44	39.15	3.62	63.35	**66.16**	**+2.81**
Qwen3.6-35B-A3B	10%	81.17	**81.66**	81.27	80.86	79.18	81.41	81.41	+0.00
	20%	81.17	80.51	80.99	80.59	76.91	81.46	**81.59**	+0.13
	30%	81.17	78.41	79.01	78.19	72.73	79.89	**80.31**	+0.42
	40%	81.17	74.61	74.84	75.74	66.16	78.79	**79.31**	+0.52
	50%	81.17	67.45	67.50	71.42	56.42	75.57	**76.38**	**+0.81**
	60%	81.17	54.66	53.42	69.04	32.27	70.70	**71.76**	**+1.06**
Gemma-4-26B-A4B-it	10%	81.63	80.59	80.67	80.82	79.61	81.59	**81.90**	+0.31
	20%	81.63	80.14	80.73	80.63	78.66	81.28	**81.34**	+0.06
	30%	81.63	78.19	77.97	78.42	75.90	80.15	**80.57**	+0.42
	40%	81.63	76.10	73.84	77.84	72.14	79.51	**79.89**	+0.38
	50%	81.63	71.14	68.28	38.74	30.83	77.11	**78.16**	**+1.05**
	60%	81.63	64.05	57.36	26.86	18.13	72.11	**73.36**	**+1.25**

Table 1 reports a seven-task mean for ProFiMoE that exceeds ACE in 17 of 18 model–budget settings and ties it once. The margins are largest at stronger compression. At 50% and 60% pruning, the seven-task gain over ACE is 2.83 and 2.81 points on Qwen3, 0.81 and 1.06 on Qwen3.6, and 1.05 and 1.25 on Gemma4. Figure 2 compares language-modeling quality across the same six pruning ratios on Qwen3 and Qwen3.6. We report BF16 together with NAEE, MoDES, and ACE.

Among the compressed methods shown in Figure 2, ProFiMoE achieves the lowest perplexity at every evaluated budget, with a tie with ACE at 10% on Qwen3. At 60% effective FFN pruning, ProFiMoE reduces perplexity from 10.86 to 10.18 relative to ACE on Qwen3 and from 9.98 to 9.70 on Qwen3.6.

At higher pruning ratios, full-or-zero methods remove responses from more routed experts, while ProFiMoE retains a structured Light response from each non-promoted expert. The widening difference in the reported means is consistent with the value of preserving these partial responses.

4.3 COMPONENT ABLATIONS

Table 2: Cumulative ablation of PAA and OMR on Qwen3.5-122B at 10% and 15% effective FFN pruning. Components are added from left to right. Mean accuracy is the unweighted average over six downstream accuracy tasks, excluding WikiText-2 perplexity. Bold denotes the best compressed configuration at each budget.

		10% pruning			15% pruning		
Task	Dense	ACE	+PAA	+OMR	ACE	+PAA	+OMR
PIQA ↑	94.50	94.04	**94.50**	**94.50**	93.93	94.10	**94.28**
ARC-Easy ↑	95.12	95.13	**95.14**	**95.14**	95.08	**95.11**	**95.11**
ARC-Challenge ↑	96.16	96.10	96.11	**96.16**	95.48	96.04	**96.08**
GPQA-Diamond ↑	53.54	51.14	53.34	**53.48**	48.99	53.01	**53.27**
MATH-500 ↑	87.20	86.50	87.00	**87.40**	85.50	86.40	**87.00**
HumanEval ↑	95.73	94.36	95.38	**95.49**	93.59	94.13	**95.36**
WikiText-2 PPL ↓	5.27	5.41	5.29	**5.28**	5.59	5.34	**5.31**
Mean accuracy ↑	87.04	86.21	86.91	**87.03**	85.43	86.47	**86.85**

Table 2 evaluates the contributions of PAA and OMR by adding them cumulatively to the ACE baseline. PAA provides the larger share of the recovery in both settings: mean accuracy increases from 86.21 to 86.91 at 10% pruning and from 85.43 to 86.47 at 15% pruning. This improvement is consistent with the role of Promotion Utility in reallocating Full execution toward routed experts that benefit more from promotion. Adding OMR further improves mean accuracy to 87.03 and 86.85, respectively. At 10% pruning, the complete configuration nearly recovers the Dense average (87.03 vs. 87.04). WikiText-2 perplexity follows the same cumulative trend, improving from 5.41 to 5.29 to 5.28 at 10% pruning and from 5.59 to 5.34 to 5.31 at 15%. Together, these results show that the two components play complementary roles: PAA produces the primary gain through promotion-aware allocation, while OMR provides additional recovery by calibrating the mixed-fidelity aggregate.

4.4 END-TO-END INFERENCE EFFICIENCY

Table 3 reports end-to-end latency on Qwen3-30B-A3B at 60% target effective FFN pruning using an NVIDIA A100 GPU. We measure time to first token (TTFT) and time per output token (TPOT) across input lengths and batch sizes.

Table 3: End-to-end inference efficiency on Qwen3-30B-A3B at 60% effective FFN pruning on an NVIDIA A100. Lower is better; BF16 denotes the uncompressed baseline.

		TTFT (ms) ↓		TPOT (ms/tok) ↓	
Length	Batch	BF16	ProFiMoE	BF16	ProFiMoE
256	1	143.3	**101.3**	**34.8**	35.2
512	1	188.7	**123.3**	**55.8**	61.1
1024	1	266.1	**131.1**	87.1	**70.5**
1024	2	300.9	**168.8**	96.2	**74.4**
1024	4	340.4	**198.0**	129.8	**103.8**

ProFiMoE reduces TTFT by 29.3–50.7% across settings. At length 1024, TPOT improves by 19.1%, 22.7%, and 20.0% for batch sizes 1, 2, and 4, respectively. The latency reduction becomes more evident for longer inputs and larger batch sizes, where FFN computation dominates the end-to-end cost. At length 1024, ProFiMoE improves both latency metrics across the measured batch sizes.

5 CONCLUSION

ProFiMoE recasts post-routing MoE compression as token-adaptive fidelity allocation: routed experts execute at either Full or Light fidelity, with Full computation assigned according to the marginal response recovered by promotion. The framework combines Promotion-Aware Allocation with Observed-Mass Rescaling while leaving the pretrained router and expert functions unchanged. Across three MoE architectures and six target FFN pruning ratios, the reported results favor ProFiMoE most strongly under high compression. On Qwen3 at 50% and 60% target pruning, the seven-task mean is 2.83 and 2.81 points above the ACE reference. On an NVIDIA A100, ProFiMoE achieves up to 50.7% lower TTFT and 22.7% lower TPOT at 60% target pruning. These results show the value of allocating fidelity within the routed expert set.

AI USE STATEMENT

Generative AI tools assisted with language editing, equation transcription, LaTeX preparation, and presentation checks. They were not used to generate experimental results. The authors remain responsible for the technical content, evidence, and disclosures.

REPRODUCIBILITY STATEMENT

Section 3 specifies the Full/Light execution modes, allocation rule, and rescaling equations. Appendix A gives the analytical derivations, and Appendices C and D report per-budget perplexity and task-level scores. Section 4 states the evaluated models, tasks, target FFN pruning ratios, and latency settings.

REFERENCES

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, Yun-Hsuan Sung, and Sumit Sanghai. CoLT5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.309. URL https://aclanthology.org/2023.emnlp-main.309/.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10865–10873, 2024. doi: 10.1609/aaai.v38i10.28960. URL https://ojs.aaai.org/index.php/AAAI/article/view/28960.

Sikai Bai, Haoxi Li, Jie Zhang, Zicong Hong, and Song Guo. DiEP: Adaptive mixture-of-experts compression through differentiable expert pruning. *arXiv preprint arXiv:2509.16105*, 2025. URL https://arxiv.org/abs/2509.16105.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

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. URL https://arxiv.org/abs/1803.05457.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *International Conference on Learning Representations*, pp. 79643–79672, 2025. URL https://openreview.net/forum?id=T26f9z2rEe.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.696. URL https://aclanthology.org/2024.acl-long.696/.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and Xiaojuan Qi. Mixture compressor for mixture-of-experts LLMs gains more. *arXiv preprint arXiv:2410.06270*, 2024b. URL https://arxiv.org/abs/2410.06270.

Yushi Huang, Zining Wang, Zhihang Yuan, Yifu Ding, Ruihao Gong, Jinyang Guo, Xianglong Liu, and Jun Zhang. MoDES: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.15690. CVPR 2026; update URL/pages when CVF proceedings appear.

Gagan Jain, Nidhi Hegde, Aditya Kusupati, Arsha Nagrani, Shyamal Buch, Prateek Jain, Anurag Arnab, and Sujoy Paul. Mixture of nested experts: Adaptive processing of visual tokens. In *Advances in Neural Information Processing Systems*, volume 37, pp. 58480–58497, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6b768359d0e8925164f61f381a748441-Abstract.html.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot MoE compression. *arXiv preprint arXiv:2510.13999*, 2025. URL https://arxiv.org/abs/2510.13999.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BjyvwnXXVn_.

Baihui Liu, Kaiyuan Tian, Wei Wang, Zhaoning Zhang, Linbo Qiao, and Dongsheng Li. Alloc-MoE: Budget-aware expert activation allocation for efficient mixture-of-experts inference. *arXiv preprint arXiv:2604.08133*, 2026a. URL https://arxiv.org/abs/2604.08133.

Zongfang Liu, Guangyi Chen, Shengkun Tang, Yifan Shen, Huan Wang, and Xin Yuan. AIMER: Calibration-free task-agnostic MoE expert pruning. *arXiv preprint arXiv:2603.18492*, 2026b. URL https://arxiv.org/abs/2603.18492.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6159–6172. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.334. URL https://aclanthology.org/2024.acl-long.334/.

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

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all. Technical report / model documentation, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b. Also available at https://huggingface.co/Qwen/Qwen3.6-35B-A3B.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router. *arXiv preprint arXiv:2410.12013*, 2024. URL https://arxiv.org/abs/2410.12013.

Zukang Xu, Zhixiong Zhao, Xing Hu, Jiangyong Yu, Houji Wen, Jun Li, Zhe Jiang, and Dawei Yang. ACE: Adaptive calibration-free expert skipping for MoE-based LLMs, 2026. URL https://arxiv.org/abs/2609.05228.

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. URL https://arxiv.org/abs/2505.09388.

Zijia Zhao, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-K routing: Boosting the efficiency of MoE-based LLMs. In *International Conference on Learning Representations*, pp. 89619–89635, 2025. URL https://openreview.net/forum?id=9CqkpQExe2.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022. URL https://arxiv.org/abs/2202.08906.

A Analytical Characterization

A.1 Promotion-utility residual bound

Proposition 1 (Promotion-utility residual bound)**.** *For any Full/Light mask M,*

$$\|y^*(x) - y_O(x, M)\|_2 \leq \sum_{i:M_i=0} R_i(x). \tag{25}$$

Substituting Eq. (2) into Eq. (1) leaves only Light-executed experts. Applying the triangle inequality gives

$$\|y^* - y_O\|_2 = \left\| \sum_{i:M_i=0} g_i(f_i - f_i^S) \right\|_2 \tag{26}$$

$$\leq \sum_{i:M_i=0} g_i \|f_i - f_i^S\|_2 = \sum_{i:M_i=0} R_i. \tag{27}$$

This bound does not equate the sum of Promotion Utilities with the actual output error, since residual vectors from different experts can interact or cancel. Instead, it provides an additive upper-bound decomposition of the mixed-fidelity residual and motivates Promotion Utility as a per-expert allocation signal.

A.2 Minimum equal-cost cumulative allocation

Proposition 2 (Minimum equal-cost coverage of the cumulative rule)**.** *Consider the cumulative Top-P allocation rule before the router top-1 inclusion. Assume all experts use the same Light Subnet width S, so each Light-to-Full promotion has incremental cost $d_{\text{ff}} - S$. For any nonnegative normalized promotion scores p_i, the descending cumulative Top-P prefix contains the fewest promoted experts among all subsets whose total promotion mass is at least P_B.*

Let $p_{(1)} \geq \cdots \geq p_{(k)} \geq 0$. Among all sets of cardinality m, the descending prefix maximizes total promotion mass because replacing any selected lower-ranked element with an unselected higher-ranked element cannot reduce the sum. If the shortest prefix reaching P_B has size m^*, no subset with fewer than m^* experts can reach the same threshold. Since every promotion has equal incremental cost $d_{\text{ff}} - S$, the prefix also minimizes incremental channel-equivalent cost. The deployed allocation additionally forces the router's top-1 expert to execute in Full. If that expert is not already in the cumulative prefix, the top-1 inclusion adds one Full Expert beyond the minimum-cardinality solution characterized above.

A.3 Error propagation in the Promotion-utility proxy

For analysis, define the unstabilized Fidelity Gap

$$\eta_i^\star(x) = \frac{\|f_i(x) - f_i^S(x)\|_2}{\|f_i(x)\|_2}, \qquad \|f_i(x)\|_2 > 0. \tag{28}$$

Then $R_i(x) = C_i(x)\eta_i^\star(x)$ exactly. The implementation uses the stabilized denominator $\|f_i(x)\|_2 + \epsilon$ in Eq. (6) for numerical robustness; the analysis below uses $\eta_i^\star(x)$ to expose the exact multiplicative structure.

Let $\widehat{C}_i(x) = g_i(x) a_i^{\text{ACE}}$ denote the estimated Full Contribution and let $\widehat{\eta}_i = \widetilde{h}_i$ denote the corresponding pre-execution Fidelity-Gap estimate. Thus, $\widehat{R}_i^{\text{PAA}}(x) = \widehat{C}_i(x)\widehat{\eta}_i$. Suppose their relative errors satisfy

$$\begin{aligned} (1-\delta_C)C_i(x) \leq \widehat{C}_i(x) \leq (1+\delta_C)C_i(x), \\ (1-\delta_\eta)\eta_i^\star(x) \leq \widehat{\eta}_i \leq (1+\delta_\eta)\eta_i^\star(x), \end{aligned} \tag{29}$$

where $\delta_C, \delta_\eta \in [0, 1)$. Multiplying the two bounds gives

$$(1-\delta_C)(1-\delta_\eta)R_i(x) \leq \widehat{R}_i^{\text{PAA}}(x) \leq (1+\delta_C)(1+\delta_\eta)R_i(x). \tag{30}$$

This bound characterizes error propagation through the product: the quality of the Promotion Utility proxy depends directly on the quality of its contribution and Fidelity-Gap estimates. The decomposition motivates separate correlation and heterogeneity diagnostics for the two factors.

A.4 OMR as a first-order scalar correction

Define the residual mismatch of the proportional observation model as

$$\xi(x) = y_O(x) - m_O(x)y^*(x). \tag{31}$$

When $m_O(x) > \epsilon_m$, Eq. (24) gives

$$\widehat{y}(x) - y^*(x) = \frac{\xi(x)}{m_O(x)}, \qquad \|\widehat{y}(x) - y^*(x)\|_2 = \frac{\|\xi(x)\|_2}{m_O(x)}. \tag{32}$$

Thus, OMR removes the scalar attenuation represented by the estimated Observed Mass; mismatch not captured by this scalar model remains in $\xi(x)$.

For $y_O(x) \neq 0$, the oracle scalar minimizing $\|\beta y_O(x) - y^*(x)\|_2$ is

$$\beta^*(x) = \frac{\langle y^*(x), y_O(x)\rangle}{\|y_O(x)\|_2^2}. \tag{33}$$

When the mixed-fidelity response is well approximated by scalar attenuation, $y_O(x) \approx m_O(x)y^*(x)$, the OMR factor $1/m_O(x)$ approaches this oracle scalar. OMR is therefore a first-order scale correction rather than a general reconstruction of the Full-execution output.

B Connection to ACE

ACE (Xu et al., 2026) supplies the contribution-estimation lineage for ProFiMoE. Its Global Spectral Proxy (GSP) estimates each expert's response capacity from the frozen SwiGLU gate, up, and down projections together with RMSNorm scaling. Router-Conditioned Refinement (RCR) constructs an expert-specific direction from centered router weights and evaluates the corresponding expert on that direction. Both quantities are computed before inference and stored as expert-indexed tables. For each token, ACE combines each table with the current router gates, normalizes the two score sets separately over the routed top-k experts, and skips a slot only when both views assign it a low score. The retained gates are then renormalized.

ProFiMoE uses a different execution decision. ACE selects which routed experts execute; PAA selects which routed experts receive Full execution while the others execute their Light Subnets. The ACE-derived contribution prior in Eq. (17) captures the potential size of a Full response. The Fidelity Gap in Eq. (18) measures how much of that response a Light Subnet is expected to miss. Their product therefore targets the incremental value of Full execution rather than the value of retaining an expert at all. After execution, OMR rescales an aggregate containing both Full and Light responses, whereas ACE renormalizes the gates of the retained Full experts.

C WikiText-2 Perplexity by Budget

Table 4 gives the numerical values underlying the ProFiMoE curves in Figure 2. Perplexity increases with the pruning ratio on both evaluated models. From 10% to 60%, it changes from 7.55 to 10.18 on Qwen3 and from 7.14 to 9.70 on Qwen3.6. WikiText-2 is reported separately from the seven-task downstream average.

Table 4: WikiText-2 perplexity of ProFiMoE at sequence length 2048. Lower is better.

Model	10%	20%	30%	40%	50%	60%
Qwen3-30B-A3B	7.55	7.63	7.78	8.06	8.61	10.18
Qwen3.6-35B-A3B	7.14	7.30	7.52	7.83	8.37	9.70

D TASK-LEVEL RESULTS

Tables 5–7 expand the seven-task means in Table 1 into individual benchmark scores. Scores for PIQA, ARC-Easy, ARC-Challenge, MATH-500, and GPQA-Diamond are accuracies; HumanEval and LiveCodeBench are pass@1. All entries are percentages, and the mean gives each task equal weight. The appendix tables report ProFiMoE at the same six target effective FFN pruning ratios as the main comparison.

The task-level results expose different responses to increasing pruning. On Qwen3, ARC-Easy changes from 94.32 at 10% pruning to 90.19 at 60%, while GPQA-Diamond changes from 57.58 to 31.31 and LiveCodeBench from 54.18 to 32.77. On Gemma-4, ARC-Easy remains above 93 across the six budgets, whereas MATH-500 declines from 88.80 to 70.60 and LiveCodeBench from 59.21 to 36.81. The main-table mean therefore summarizes task trajectories with different sensitivity to the Full/Light budget.

Table 5: Task-level performance of ProFiMoE on Qwen3-30B-A3B-Instruct-2507 across six target effective FFN pruning ratios. Mean is the unweighted average of the seven downstream tasks; ARC-Easy and ARC-Challenge are separate tasks.

Budget	PIQA	ARC-E	ARC-C	MATH-500	GPQA-D	HumanEval	LiveCodeBench	Mean
10%	91.33	94.32	93.09	79.80	57.58	95.12	54.18	80.77
20%	90.48	94.40	93.09	79.60	58.08	93.90	53.98	80.50
30%	89.39	94.36	92.92	78.60	56.57	93.90	52.78	79.79
40%	88.68	94.15	92.41	78.60	52.02	92.07	51.58	78.50
50%	87.49	93.98	91.64	74.60	53.54	90.85	47.78	77.13
60%	82.48	90.19	88.57	62.80	31.31	75.00	32.77	66.16

Table 6: Task-level performance of ProFiMoE on Qwen3.6-35B-A3B across six target effective FFN pruning ratios. Mean is the unweighted average of the seven downstream tasks; ARC-Easy and ARC-Challenge are separate tasks.

Budget	PIQA	ARC-E	ARC-C	MATH-500	GPQA-D	HumanEval	LiveCodeBench	Mean
10%	94.45	94.87	95.31	86.00	46.46	96.34	56.47	81.41
20%	94.45	94.87	95.39	85.00	47.98	95.73	57.68	81.59
30%	93.31	94.74	95.14	84.40	46.96	94.51	53.11	80.31
40%	93.36	94.61	95.05	82.40	43.94	95.73	50.07	79.31
50%	92.76	94.49	94.37	81.60	36.36	93.90	41.18	76.38
60%	91.15	94.13	93.86	74.20	33.81	84.15	31.05	71.76

Table 7: Task-level performance of ProFiMoE on Gemma-4-26B-A4B-it across six target effective FFN pruning ratios. Mean is the unweighted average of the seven downstream tasks; ARC-Easy and ARC-Challenge are separate tasks.

Budget	PIQA	ARC-E	ARC-C	MATH-500	GPQA-D	HumanEval	LiveCodeBench	Mean
10%	87.27	94.98	94.91	88.80	53.01	95.12	59.21	81.90
20%	87.27	94.15	94.11	86.80	53.54	95.73	57.77	81.34
30%	87.05	94.19	94.54	83.80	53.25	96.34	54.80	80.57
40%	86.29	94.45	94.54	82.80	51.87	95.10	54.15	79.89
50%	85.36	94.47	94.21	80.40	50.04	93.75	48.89	78.16
60%	84.82	93.85	92.45	70.60	43.40	91.56	36.81	73.36