FRAGILITY-AWARE ADAPTIVE ROUTING FOR QUANTIZED MIXTURE-OF-EXPERTS SERVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantizing the experts of a Mixture-of-Experts (MoE) model is standard practice for relieving memory pressure during serving, producing a pool of pre-quantized model copies: *instances*, whose quality cost varies sharply across requests. Existing systems have two limitations: statically committing to a single instance for the workload and relying on ad hoc heuristic request score predictors. Closing this gap, we propose an adaptive routing framework with theoretical guarantees that assigns each incoming request to an instance so as to maximize throughput within a quality-degradation budget. Underlying the framework is a structural result: a small-noise expansion factors quality loss over experts into request-independent *expert fragility* and request-dependent *expert affinity*. This factorization yields FWP (Fragility-Weighted Perplexity), a lightweight quality estimator read from a single prefill pass on the cheapest instance and transferred to every other instance by affine calibration. Building on FWP, we cast routing as a window-level linear program and prove that the induced per-request greedy policy is KKT-consistent with the LP optimum. Empirically, on complete Qwen3-30B-A3B instances, FWP-guided allocation at a fixed quality budget comes within 1.5% of an allocation that knows each request's true quality loss, well ahead of activation-frequency and prompt-length signals. Turning this signal into live serving, the LP-derived router over 4-bit, 4/8-bit, and 8-bit instances meets every per-class quality budget at up to $1.20\times$ the measured throughput of the only static deployment that satisfies all classes.

1 INTRODUCTION

Mixture-of-Experts (MoE) architectures are becoming the modern paradigm for large language models, delivering higher throughput than dense models at comparable quality (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024; Qwen Team, 2025). Under serving, however, GPU memory bottlenecks this throughput, and post-training expert quantization is the standard practice for relieving it (Frantar et al., 2023; Lin et al., 2024; Badri & Shaji, 2023; Frantar & Alistarh, 2024; Kim et al., 2023). Quantization thus creates a trade-off between quality and throughput: lower bitwidths raise decoding throughput but degrade output quality. Moreover, this quality cost varies sharply across requests, because some requests activate experts that quantize gracefully, while others rely on experts prone to sharp degradation at low bitwidths. Serving a quantized MoE model therefore means choosing a point on this trade-off, and a static deployment commits every request to the same point.

We study this trade-off over a fixed pool $\mathcal{A}$ of pre-quantized copies of the same MoE base, each at a distinct quantization configuration and held resident at serving time; we call these copies *instances*. Such a pool need not add memory, since high-traffic services already run several data-parallel replicas of one model, and quantizing some replicas more aggressively only shrinks their footprint. Because instances are provisioning inputs, our routing composes with any quantizer or bit-allocation method that produces them. This raises the central question of this paper:

How should a serving system route each incoming request among the pool's instances to maximize throughput under a per-class quality-degradation budget?

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.

Existing approaches have two limitations. Some pick a single instance for the entire workload, ignoring request-level heterogeneity, while others use hand-crafted per-request scorers that must be re-tuned whenever the workload shifts. Cross-model routing (Chen et al., 2024; Ong et al., 2025; Hu et al., 2024; Ding et al., 2024) is closer in spirit, yet it selects different models rather than bitwidth instances of one model.

Routing can be principled because quantization loss has a simple structure. **Expert fragility** is an intrinsic per-expert property: how an expert's quantization response degrades at low bitwidths (Chowdhury et al., 2026). In contrast, **expert affinity** is request-dependent: the gating weight with which a request activates each expert (Shazeer et al., 2017; Fedus et al., 2022). Under a small-noise expansion in a two-expert sparse-MoE model (Section 3.1), we prove that quality loss factors over experts into request-independent expert fragility times request-dependent expert affinity. So far, expert-quantization methods use fragility only for offline bit allocation (Chowdhury et al., 2026). We instead combine it with affinity into an online per-request signal, **FWP (Fragility-Weighted Perplexity)**: token-level negative log-likelihood weighted by the inverse mean router norm of the activated experts. This weighting amplifies tokens routed to fragile experts. Computing FWP needs one prefill pass on the cheapest instance, and offline affine maps convert this measurement into quality estimates for every other instance. As a result, the router needs no per-candidate scorer.

Building on FWP, we cast routing as a **window-level linear program** over a sliding batch of recent requests. The program maximizes served throughput subject to per-instance capacity and per-class quality budgets, with each class grouping requests of a shared quality tolerance. From its dual, we derive a per-request score combining throughput gain, the FWP quality estimate, and dual prices on capacity and quality. We prove that this score is **KKT-consistent with the LP optimum** (Section 3.3): under optimal prices, sending each request to its highest-scoring instance matches the LP optimum. Routing thus reduces to one argmax per request, without re-solving the LP at every arrival.

Experiments on Qwen3-30B-A3B and DeepSeek-V2-Lite support each layer of this framework, from the offline signal to live serving (Section 4). Offline, the reduced-form surrogate ranks 65 held-out quantization configurations with up to 0.822 pairwise accuracy. At the request level, FWP read from the cheapest complete Qwen3-30B-A3B instance ranks which requests degrade most on the next instance with top-20% AUROC 0.75. At a fixed quality budget, FWP-guided routing reaches $1.284\times$ the decode throughput of the highest-bit instance under the memory-bound decode model. This gain lies within 1.5% of routing on the true per-request loss and well ahead of routing on activation frequency or prompt length. Our main result comes from live serving on three GPUs, where the LP-derived router serves mixed quality classes at $1.06\times$ to $1.20\times$ the throughput of the only static deployment meeting every class budget, and its throughput rises monotonically as the budget loosens. As an extension under prefill/decode disaggregation (Zhong et al., 2024; Patel et al., 2024), one or two additional prefills on other instances further sharpen routing, with the best prefill count set by the model's active-parameter weight.

Our contributions are:

1. **A theoretical framework for quality-aware routing in quantized MoE serving.** We prove that quality loss admits an exact expert-wise decomposition (Appendix Proposition B.1) and, under a small-noise expansion, factors over experts into request-independent expert fragility times request-dependent expert affinity. We then cast routing as a window-level linear program and prove that its dual induces a per-request greedy score that is KKT-consistent with the LP optimum.

2. **FWP (Fragility-Weighted Perplexity), an online quality observable.** A single prefill pass on the cheapest instance yields a fragility-weighted perplexity that transfers to every other instance via affine calibration, so one pass scores the whole pool with no per-workload scorer redesign.

3. **An adaptive routing policy, tested in live serving.** The policy routes each request by the argmax of the LP-derived score, pricing instance capacity and each class's quality budget against the calibrated FWP estimate, so it adapts to load and class mix through its prices without re-solving the LP or re-provisioning the pool. Deployed on a production serving stack with 4-bit, 4/8-bit, and 8-bit instances, it serves mixed quality classes at up to $1.20\times$ the measured throughput of the only static deployment that meets every class budget on the same GPUs, and its throughput rises monotonically as the budget loosens.

2 Background and Problem Setting

2.1 Quantized MoE Serving as a Request-Routing Problem

Notation. We consider an autoregressive MoE model with L sparse layers. At token t, layer ℓ routes the hidden state h_t through a top-k router with gate weights $g_{t,e}$ to an expert subset $\mathcal{K}_t \subseteq \mathcal{E}_\ell$, producing $y_t = \sum_{e \in \mathcal{K}_t} g_{t,e} f_e(h_t)$. In our analysis only the expert weights $(W_{1,e}, W_{2,e})$ are quantized and the router stays in BF16; a quantization configuration assigns bitwidth b_e to each expert e. This asymmetry is the architectural lever: routing is observable in real time, while quantization damage concentrates on a small, request-dependent subset of experts. For a request x with token set $\mathcal{T}(x)$, per-expert affinity is

$$\rho_e(x) = \frac{1}{|\mathcal{T}(x)|} \sum_{t \in \mathcal{T}(x)} g_{t,e}, \tag{1}$$

turning a per-expert quantization choice into a per-request quality risk.

Pool and workload. At serving time, a gateway dispatches each whole request to one of a fixed pool $\mathcal{A} = \{1, \ldots, A\}$ of resident pre-quantized model copies, with quantization configurations $\mathbf{b}^{(a)}$ and replica counts fixed by provisioning; production gateways already expose such choices, for example provider selection with quantization filters and external routing across data-parallel endpoints (OpenRouter, 2026; vLLM Team, 2026). Requests arrive as a workload trace, each carrying a class g_t with expected output-token count L_{g_t}, profiled TTFT/TPOT latency, and a quality tolerance τ_{g_t}. A policy π maps state and request to an action $a_t \in \mathcal{A}$, which is the routing decision. The objective is to maximize served throughput subject to per-class expected quality degradation staying within budget τ_g, formalized as the window-level LP in Section 3.3.

2.2 Related Work

Our work sits at the junction of four lines. First, sparse MoE models (from gated experts (Shazeer et al., 2017) through GShard and Switch Transformers (Lepikhin et al., 2021; Fedus et al., 2022) to Mixtral, DeepSeekMoE, and Qwen MoE (Jiang et al., 2024; Dai et al., 2024; Qwen Team, 2024; 2025)) supply the architectural substrate but do not address how heterogeneously quantized instances should be served. Second, post-training quantization (Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023; Badri & Shaji, 2023) and MoE-specific compressors (Frantar & Alistarh, 2024; Kim et al., 2023; Chowdhury et al., 2026) produce the fixed pool of low-bit instances we consume; the per-expert router-norm fragility indicator of Chowdhury et al. (2026) in particular motivates the request-level lift that yields our FWP signal. Third, serving systems (Yu et al., 2022; Kwon et al., 2023; Zhong et al., 2024; Sun et al., 2024) optimize throughput and latency without an explicit quantization-quality risk term, which our routing policy adds. Fourth, cost-aware request routing across heterogeneous models (Chen et al., 2024; Ong et al., 2025; Hu et al., 2024; Ding et al., 2024) is the closest spirit, but those routers select *which model*, not *which bitwidth instance* of the same model, and lack the expert-fragility lens. An expanded discussion, including the link to selective prediction, appears in Appendix A.

3 Theoretical Framework

This section builds the theory of the routing layer in three steps, each of which fixes an implementation choice. The factorization tells the system *what* to compute online and what to compute offline: expert affinity is read from the router when a request arrives, while expert fragility is measured once in advance. The FWP amplification result tells it *how* to read the request-level signal, from a single prefill pass. The KKT-consistent score tells it *when* to update: dual prices change once per control window, and each request is routed by a single argmax.

3.1 Expert Fragility and the Reduced-Form Surrogate

We begin with an explicit model that isolates the structure we want to exploit (full derivation in Appendix B).

Two-expert sparse-MoE model. Consider a sparse top-1 MoE classifier with two experts, e_r and e_c. Each request belongs to a rare population (mass $\alpha < \frac{1}{4}$) routed to e_r, or a common population (mass $1-\alpha$) routed to e_c. The router is deterministic: $g(x) = e_r$ if $X = r$, else e_c. Let $m_j > 0$ be the full-precision signed classification margin for population $j \in \{r, c\}$, and assume $0 < m_r < m_c$: the rare expert has a weaker activation margin, reflecting the fragility mechanism in the literature (Chowdhury et al., 2026). The prediction loss is logistic, $\ell(z) = \log(1 + e^{-z})$, and full-precision quality is $Q_{\text{full}} := -\mathbb{E}[\ell(m_X)]$.

When each expert e_j is quantized to b_j bits, the margin is perturbed by a zero-mean random variable $\xi_j(b_j)$ with variance scaling $v_j 2^{-2b_j}$ (the standard uniform-quantization scaling). The quantized quality is $Q(\mathbf{b}) := -\mathbb{E}[\ell(m_X + \xi_X(b_X))]$, and the quality loss is $\Delta Q(\mathbf{b}) := Q_{\text{full}} - Q(\mathbf{b})$.

Exact affinity and fragility decomposition. Conditioning on the request type gives the exact decomposition

$$\Delta Q(\mathbf{b}) = \alpha\, \Delta_r(b_r) + (1-\alpha)\, \Delta_c(b_c), \tag{2}$$

where $\Delta_j(b_j) := \mathbb{E}[\ell(m_j + \xi_j(b_j)) - \ell(m_j)]$. This cleanly factors the loss into two objects: the coefficients α and $1-\alpha$ are *affinity* weights (how often each expert is used), while Δ_r and Δ_c are *intrinsic fragility* functions (how much damage quantization does to each expert). These two quantities should not be combined, and the routing score we derive in Section 3.3 inherits both.

Small-noise expansion and reduced form. To obtain a practical parameterization, we expand each $\Delta_j(b_j)$ around the full-precision margin. Under the variance scaling above, a second-order Taylor expansion yields the dominant term proportional to 2^{-2b_j} (Appendix B, Lemma B.1). More generally, linearizing the quantization response near zero noise gives a separable form with a coefficient multiplied by a bit-decay function.

Theorem 3.1 (Reduced-form surrogate)**.** Under the two-expert model and the small-noise expansion,

$$\Delta Q(\mathbf{b}) = \alpha\, c_r\, \phi(b_r) + (1-\alpha)\, c_c\, \phi(b_c) + r(\mathbf{b}), \tag{3}$$

where $c_j = \frac{1}{2}\ell''(m_j)v_j > 0$ is an expert-specific fragility coefficient, $\phi(b) = 2^{-2b}$ is the bit-decay function, and the remainder satisfies $|r(\mathbf{b})| \le \kappa \sum_j 2^{-3b_j}$ for a constant κ depending on the local curvature of ℓ.

Theorem 3.1 is the constructive justification for the empirical surrogate we fit. The fragility coefficient c_j is controlled by a monotone decreasing upper bound in the expert's router-derived margin proxy $\widetilde{\Lambda}_j$ (Appendix B, Proposition B.3), formalizing a conservative connection between router-side geometry and quantization fragility.

Empirical surrogate and request-level lift. Guided by Theorem 3.1, the configuration-level model we fit to real perturbation data is

$$\widehat{\Delta Q}_{\text{cfg}}(\mathbf{b}) = \sum_{e \in \mathcal{E}} \widehat{c}_e^{\,\text{free}}\, \phi(b_e), \tag{4}$$

where $\widehat{c}_e^{\,\text{free}}$ is calibrated by leave-one-expert-out perturbations and $\phi(b) \in \{2^{-b}, 2^{-2b}\}$ is selected empirically using the centered- and biased-noise expansions (Lemma B.2). As a restricted ablation we also fit a structured family parameterized by router norm and MaxVar (Appendix B); both families are evaluated in Section 4.

The configuration-level form lifts to the request level via the affinity weights from Eq. (1):

$$\widehat{\Delta Q}_{\text{req}}(x, \mathbf{b}) = \sum_{e \in \mathcal{E}} \rho_e(x)\, \widehat{c}_e\, \phi(b_e), \tag{5}$$

with $\widehat{c}_e$ from either family. The lift inherits the two roles of the exact decomposition in Eq. (2): *affinity* through $\rho_e(x)$ (read from the router's gates at decision time), and *fragility* through $\widehat{c}_e$ (precomputed once). The lift extends the two-expert additive form to top-k multi-layer MoE under gate-normalized aggregation, and Section 4 tests this multiplicative $\rho_e \cdot \widehat{c}_e \cdot \phi(b_e)$ form directly. A direct consequence of Eq. (5) is that for two instances $\mathbf{b}, \mathbf{b}'$ over the same expert set, the lifted surrogates differ only through the bit-decay ratios $\phi(b_e)/\phi(b'_e)$. Cross-instance prediction therefore reduces to an *affine* recalibration that absorbs per-instance scale and offset, which we exploit empirically in Section 4.2.

3.2 FWP: FRAGILITY-WEIGHTED PERPLEXITY

The lifted surrogate in Eq. (5) is a metric-agnostic predictor. The routing gain, however, depends on how strongly the quality observable separates fragile-expert degradation from bulk-token noise. We now construct a concrete observable, FWP, and prove that it amplifies the signal of interest.

Design. We design FWP from a per-expert fragility signal, the router ℓ_2 norm. Fragility has been tied to the router-norm change during training (Chowdhury et al., 2026); we use the final norm, which is readable from any released checkpoint without training logs. Its link to the fragility coefficient is established through the router-margin bound in Section 3.1. We lift this expert-level signal to a per-request, per-token observable by reading the router norm at the token where it acts. For a request x with token sequence $(x_1, \ldots, x_{|x|})$, let $\bar{\Lambda}_i(x)$ be the average router norm of the experts activated by token x_i. A small $\bar{\Lambda}_i(x)$ flags a token routed to fragile experts, where the NLL under quantization deviates most. Uniform NLL averaging dilutes this signal, while inverse-router-norm weighting emphasizes it. This yields the FWP (Fragility-Weighted Perplexity) observable. Define normalized token weights

$$w_i(x) = \frac{(\bar{\Lambda}_i(x) + \epsilon_w)^{-1}}{\frac{1}{|x|}\sum_{j=1}^{|x|}(\bar{\Lambda}_j(x) + \epsilon_w)^{-1}}, \tag{6}$$

with $\epsilon_w > 0$, and the log-FWP observable

$$\ell_{\mathrm{FWP}}(x; \mathbf{b}) = \frac{1}{|x|}\sum_{i=1}^{|x|} w_i(x)\big(-\log p_{\mathbf{b}}(x_i \mid x_{<i})\big). \tag{7}$$

Three properties matter: (i) small router norm ⇒ fragile expert ⇒ large weight, inheriting the per-expert fragility convention; (ii) per-token weighting makes FWP request-dependent even for the same instance; (iii) the log form keeps FWP on the NLL-delta scale for direct calibration. For routing, we use the request-level FWP increase over the full-precision reference $\ell_{\mathrm{FWP}}(x; \mathbf{b}_{\mathrm{full}})$, which at serving time is predicted from prefill-observable features (Section 4.2), so no full-precision model is invoked online:

$$\Delta\ell_{\mathrm{FWP}}(x; \mathbf{b}) = \ell_{\mathrm{FWP}}(x; \mathbf{b}) - \ell_{\mathrm{FWP}}(x; \mathbf{b}_{\mathrm{full}}), \tag{8}$$

which the lifted surrogate approximates as $\widehat{\Delta\ell}_{\mathrm{FWP}}(x; \mathbf{b}) \approx \widehat{\Delta Q}_{\mathrm{req}}(x, \mathbf{b})$ when $Q(\mathbf{b})$ in Eq. (5) is instantiated as log-FWP.

FWP amplifies fragile-expert degradation. Under the margin ordering $m_r < m_c$ and variance condition $v_r \geq v_c$ from Section 3.1, the rare expert is intrinsically more fragile than the common expert (Appendix Proposition B.2), so $\Delta_r(b) \geq \Delta_c(b)$ for equal bitwidths. We now prove that FWP-style inverse-router-norm weighting amplifies this gap. Under the two-expert model each request is routed to exactly one expert, so the token-level weights collapse to population-level constants $\omega_r > 1 > \omega_c$ when $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$, satisfying $\alpha\omega_r + (1-\alpha)\omega_c = 1$ (Appendix B). The two-expert FWP loss is $\Delta\ell^{\mathrm{model}}_{\mathrm{FWP}}(\mathbf{b}) = \alpha\omega_r\Delta_r(b_r) + (1-\alpha)\omega_c\Delta_c(b_c)$, while ordinary average NLL loss is $\Delta Q(\mathbf{b}) = \alpha\Delta_r(b_r) + (1-\alpha)\Delta_c(b_c)$.

Proposition 3.1 (FWP amplifies fragile-expert degradation)**.** Suppose $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$ and $\Delta_r(b_r) \geq \Delta_c(b_c)$. Then

$$\Delta\ell^{\mathrm{model}}_{\mathrm{FWP}}(\mathbf{b}) - \Delta Q(\mathbf{b}) = \alpha(\omega_r - 1)\big(\Delta_r(b_r) - \Delta_c(b_c)\big) \geq 0, \tag{9}$$

with strict positivity whenever $\Delta_r(b_r) > \Delta_c(b_c)$.

Proposition 3.1 is the theoretical basis for using FWP as the online observable. The proposition shows that FWP-style weighting widens the measured gap $\Delta_r(b_r) - \Delta_c(b_c)$ between the rare expert's degradation and the common expert's, which strengthens the routing signal. In the two-expert model every token of a request reaches the same expert, so the proposition analyzes weights normalized across the request population and certifies the inverse-fragility *rescaling* mechanism. The deployed per-request normalization in Eq. (6) adds within-request contrast when a request's tokens reach experts of different fragility, which Section 4.2 evaluates empirically. The amplification result also justifies reading a single prefill pass at decision time instead of enumerating per-expert affinity for every candidate.

3.3 LP-Based Adaptive Routing

We now use the quality estimates from Sections 3.1 and 3.2 to derive a routing policy. We formulate adaptive routing as a window-level linear program and obtain the per-request score from its dual.

Window-level LP. The serving pool $\mathcal{A}$, its quantization configurations, and its replica counts are fixed. Requests may be grouped into classes $g \in \mathcal{G}$ by service tier, prompt length, task type, or affinity pattern. Let L_g be the expected output-token count, $s_{ga}(t)$ the profiled service cost (from TTFT/TPOT), $B_g(t)$ the class demand in window t, $C_a(t)$ the capacity of instance a, and τ_g the per-class quality tolerance. The LP selects class-instance flows $x_{ga}(t) \geq 0$ to maximize served tokens subject to demand, capacity, and quality-risk constraints:

$$\max_{x \geq 0} \sum_{g,a} L_g x_{ga}(t) \;\text{ s.t. }\; \sum_a x_{ga} \leq B_g,\; \sum_g s_{ga} x_{ga} \leq C_a,\; \sum_a (\widehat{\ell}_{ga} - \tau_g) x_{ga} \leq 0, \tag{10}$$

where $\widehat{\ell}_{ga}(t) = \sum_e \rho_{g,e}(t)\, \widehat{c}_e\, \phi(b_e^{(a)})$ is the predicted quality risk (replaced by calibrated FWP measurements when the information regime permits). The quality constraint treats per-class expected degradation as linear in the allocation fraction (a first-order approximation; see Appendix C for the dual program, linearization justification, and quota tracking).

Dual structure and per-request score. Let $\pi_a \geq 0$ be the price of capacity on instance a, and $\nu_g \geq 0$ the price of quality risk for class g. The Lagrangian of Eq. (10) yields the one-step reduced reward

$$R_{ga}(t) = L_g - \pi_a(t) s_{ga}(t) - \nu_g(t)\big(\widehat{\ell}_{ga}(t) - \tau_g\big), \tag{11}$$

up to a class-specific constant. Adding a quota-deficit tracking term $d_{ga}(t)$ gives the template score

$$S_{ga}(t) = \beta d_{ga}(t) + L_g - \pi_a(t) s_{ga}(t) - \nu_g(t)\big(\widehat{\ell}_{ga}(t) - \tau_g\big), \tag{12}$$

where $\beta \geq 0$ controls how tightly the policy follows a window-level target allocation. Clipping the quality term at zero gives a variant that ignores risk below tolerance; Theorem 3.2 covers the signed score above, which is the score our live router runs (Section 4.4).

Theorem 3.2 (KKT-consistency of the routing score)**.** Assume stationary workload statistics within a control window, exact service-cost estimates, and exact class-level loss predictions, and that the projected subgradient updates for (π, ν) converge to an optimal dual solution $(\pi^\star, \nu^\star)$ of Eq. (10) (a standard property under square-summable, non-summable step sizes for convex LP duals). Then the greedy routing rule $a_t^*(g) = \arg\max_a S_{ga}(t)$ is KKT-consistent with the window-level LP optimum, that is, under optimal dual prices every instance the rule selects satisfies the LP's stationarity condition. In particular, when $\beta = 0$, the rule coincides with the greedy primal step of the Lagrangian relaxation (Appendix C, Theorem C.2).

Theorem 3.2 provides the guarantee that the per-request score is not an ad hoc heuristic: it inherits its structure from the optimal resource-allocation objective, and the three terms in Eq. (12) map directly to the three constraints in Eq. (10): throughput reward from the objective, capacity price from the instance-capacity constraint, and quality-risk price from the per-class quality constraint. The KKT result makes the policy efficient at runtime: routing reduces to a per-request argmax,

$$a_t^*(g) = \arg\max_{a \in \mathcal{A}} S_{ga}(t), \tag{13}$$

with dual prices updated once per control window rather than per arrival. In deployment the per-request overhead is one prefill pass on the cheapest instance, which supplies the FWP measurement, plus an argmax over $|\mathcal{A}|$ scores (pseudocode in Figure 5, Appendix D).

4 Results and Discussion

FWP-guided routing turns expert fragility into measured serving throughput, from the offline signal to live serving. Offline, the configuration prior preserves rank order across architectures (§4.1), and FWP beats cheap baselines on Qwen3-30B-A3B and DeepSeek-V2-Lite and transfers across instances (§4.2). With this signal, FWP-guided allocation at a fixed quality budget comes within 1.5% of routing on the true per-request loss (§4.3). Our main result follows in live serving, where the

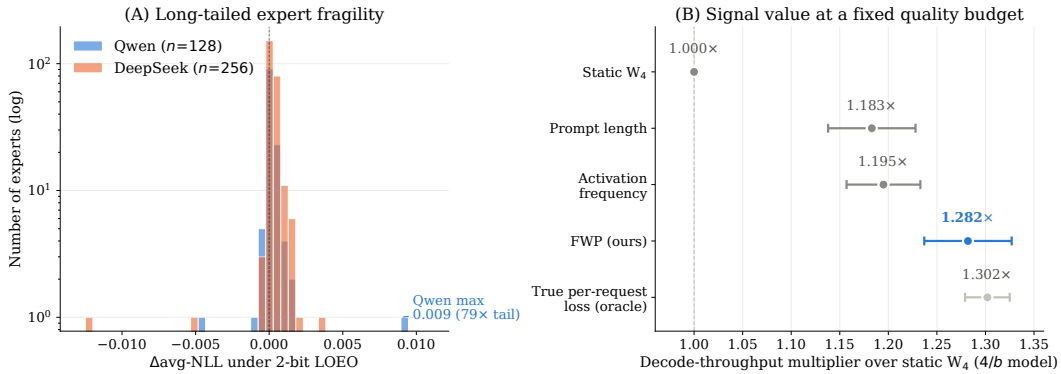


Figure 1: (A) 2-bit LOEO Δavg-NLL distribution, both architectures overlaid (sampled-expert protocol). (B) Decode-throughput multiplier over static W_4 at the fixed quality budget of Section 4.3, by the signal that orders requests (complete Qwen3-30B-A3B instances; held-out mean $\pm$ s.d. over 20 calibration/test splits).

LP-derived router serves mixed quality classes at $1.06\times$ to $1.20\times$ the throughput of the only static deployment that meets every class budget (§4.4). As an extension, extra prefills under prefill/decode disaggregation reach a utility frontier that no other multi-instance policy in our evaluation matches (§4.5).

Setup. Our evaluation proceeds in stages, from the offline signal to live serving (full protocol in Appendix D). The first stages study the signal on Qwen3-30B-A3B and DeepSeek-V2-Lite, quantizing a sample of experts with HQQ and keeping the rest at BF16 to isolate each expert's contribution: leave-one-expert-out (LOEO) perturbations at 2 to 8 bits measure expert fragility, a held-out set of 65 quantization configurations tests whether the fitted surrogate ranks them correctly, and requests from WikiText-2, OpenBookQA, MBPP, and CNN/DailyMail test FWP as a request-level signal. The next stage quantizes every expert of Qwen3-30B-A3B into complete 2-, 3-, and 4-bit instances and asks how far a fixed quality budget stretches under FWP-guided allocation. The final stage, our main result, serves complete 4-bit, 4/8-bit, and 8-bit instances live with vLLM and measures throughput and per-class quality end to end (Section 4.4).

4.1 Expert Fragility and Configuration-Level Priors

HQQ quantized leave-one-expert-out perturbations resolve low-bit expert fragility into a long-tailed effect rather than a uniform property (Figure 1A): most sampled experts contribute negligible quality loss at 2-bit, while a small minority degrade quality by orders of magnitude more than the typical expert. This concentration justifies the per-request unit of decision from §3: a per-request signal is only useful when fragility varies across requests, and the long tail confirms that variation. Cheap router-side predictors localize this tail without reproducing it. Activation frequency is the strongest of the cheap signals, yet the most-activated experts and the most fragile ones overlap on only about half of the top-32 list, so fragility ranking requires a quantization-aware probe rather than activation statistics alone. Cheap router-side signals therefore serve as the offline prior, and FWP supplies the online decision signal.

The reduced-form surrogate from §3 preserves rank order on 65 held-out configurations on both architectures, with pairwise accuracy 0.822 on DeepSeek-V2-Lite and 0.596 on Qwen3-30B-A3B (Appendix Table 2). Held-out R^2 on absolute scale is negative on both architectures: the surrogate ranks well, but it does not calibrate the magnitude of quality loss. We therefore use the surrogate for ranking only, and the deployable signal in §4.2 comes from the FWP family.

4.2 Request-Level FWP Signals and Heterogeneity

The FWP family is the strongest empirical signal in the routing layer. Measured on the same instance, FWP variants recover 95 to 98% of the top-20% highest-risk requests on both architectures, far ahead of the activation-frequency baseline (12 to 27%; Appendix Table 2); since FWP and the audit target share one forward pass, this certifies measurement reliability. Routing needs transfer: FWP read from the cheapest complete Qwen3-30B-A3B instance ranks which requests degrade most on the next instance with Spearman correlation 0.38 to 0.58 and top-20% AUROC 0.75 across both source modes, and this cross-instance signal drives the fixed-budget allocation and the live router in Section 4.4. Request-level heterogeneity is small in quality terms but large in throughput terms: the per-request quality gain over the static best-quality instance is an order of magnitude smaller than the LOEO tail, so FWP converts heterogeneity into throughput at a fixed quality budget rather than into quality.

For deployment without an offline reference, a ridge estimator on the features of one prefill pass recovers the full-reference $\log \mathrm{FWP}$ with held-out $R^2 = 0.999$ on both architectures, so no full-precision model runs online.

4.3 Fixed-Budget Allocation on Complete Instances

FWP-guided routing converts request-level risk into throughput at a fixed quality budget (Figure 1B; full results in Appendix Table 4). With the budget set 0.10 nats/token above the highest-bit instance's mean loss, no faster static instance is feasible, since W_3 alone exceeds the budget. Ordering requests by FWP from the cheapest instance and sending those below a cutoff to W_3 meets the budget at $1.284\times$ the modeled decode throughput of static W_4, with 85% of requests on W_3; for a single request class, such a cutoff rule is the form Eq. (12) takes at the quality price that makes the budget bind. The quality of the risk signal decides how far the budget stretches. Routing on activation frequency or prompt length reaches only $1.197\times$ and $1.186\times$, because a weak signal spends budget on requests that would have tolerated W_3, while FWP comes within 1.5% of routing on the true per-request loss ($1.303\times$). The gain replicates on held-out requests: with the cutoff fit on a calibration split, FWP reaches $1.282 \pm 0.045\times$ over 20 random calibration/test splits.

4.4 Live Serving with Mixed Quality Classes

The LP-derived router delivers measured throughput gains in live serving when requests carry different quality budgets (Table 1). We serve Qwen3-30B-A3B with vLLM on three A100-40GB GPUs, one instance each: W_4, W_8, and $\mathrm{W}_{4/8}$, which keeps 8-bit experts in the 24 most fragile layers (largest mean inverse router norm) and 8-bit attention and router weights. FWP is read from a prefill pass on a 3-bit reference instance. The predictor and warm-start prices are fit on a calibration half of 600 chat, math, code, and summarization prompts, and every run uses the disjoint test half. Quality is measured on the served instances themselves, as ΔNLL from BF16 on BF16's own greedy response. A quality class is a group of requests that share one quality budget, as in a service that offers several tiers of output quality. Each request belongs to one of three classes, strict, medium, and loose, with budgets of 0.005, 0.020, and 0.049 nats/token, or about $+0.5\%$, $+2\%$, and $+5\%$ perplexity. The router applies Eq. (12) with $\beta = 0$ and one quality price per class, enforcing each class's LP quality constraint at dispatch (Appendix E).

With the same three serving GPUs, the FWP-routed pool serves $1.06\times$ to $1.20\times$ the throughput of $\mathrm{W}_8\times 3$, the only static deployment that meets every class budget, and the gain grows with the loose share. It meets every class budget on the loose-heavy and drifting workloads and on the balanced workload within label noise, and exceeds the medium budget by 4% on the strict-heavy mix. The drifting workload changes its class mix without any re-provisioning, and the router follows it through its prices. A single-class sweep of the budget from 0.002 to 0.075 nats/token raises throughput monotonically from 2,325 to 5,087 tok/s with every target met (Figure 2; Appendix Table 5).

4.5 Multi-Prefill Routing Under Disaggregation

The sampled-expert deployment stress study grades the FWP-aware policy by prefill count (Appendix Table 3 and Figure 3C). A single prefill on the reference instance already carries most of the gain;

Table 1: Live serving on three A100-40GB GPUs. Each request belongs to a quality class with a budget on its class-mean ΔNLL (nats/token, vs. BF16): strict ($\Delta\text{NLL} \leq 0.005$), medium ($\leq 0.020$), and loose ($\leq 0.049$); class columns give the measured class mean. Static deployments replicate a designated quantized instance across all three GPUs; the FWP-routed pool runs one W_4, one $W_{4/8}$, and one W_8 instance under four class mixes (strict/medium/loose share of requests; the last drifts from 60/20/20 to 20/20/60). ✓ marks a met class budget and × an exceeded one; † within the label-noise standard error (± 0.0002).

Deployment	Mix S/M/L (%)	tok/s	vs. $W_8\times3$	Strict	Medium	Loose	Budgets met
Static $W_4\times3$	any	5,852	1.41×	0.094 ×	0.097 ×	0.093 ×	none
Static $W_{4/8}\times3$	any	4,799	1.16×	0.011 ×	0.012 ✓	0.012 ✓	medium, loose
Static $W_8\times3$	any	4,145	1.00×	0.001 ✓	0.001 ✓	0.001 ✓	all
FWP-Routed Pool	60/20/20 (strict-heavy)	4,399	1.06×	0.0033 ✓	0.0208 ×	0.0480 ✓	strict, loose
FWP-Routed Pool	33/33/33 (balanced)	4,651	1.12×	0.0043 ✓	0.0190 ✓	0.0491 ✓†	all
FWP-Routed Pool	20/20/60 (loose-heavy)	**4,957**	**1.20×**	0.0039 ✓	0.0167 ✓	0.0470 ✓	all
FWP-Routed Pool	60/20/20 → 20/20/60	4,607	1.11×	0.0024 ✓	0.0171 ✓	0.0476 ✓	all

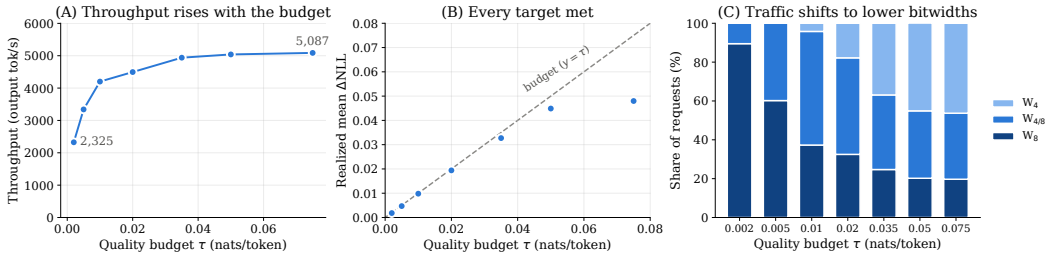


Figure 2: Single-class budget sweep on the FWP-routed pool (test half). (A) Throughput rises monotonically as the quality budget loosens. (B) The realized mean ΔNLL stays at or below every budget. (C) The router meets looser budgets by shifting requests from W_8 to $W_{4/8}$ and W_4.

two prefills on the best pair (W_2-W_3) give the best Qwen utility, and three prefills with queue-aware dispatch give the best SLO compliance on both architectures and the best utility on DeepSeek.

The multi-prefill gain depends on whether prefill and decode share hardware. Under prefill/decode disaggregation (Zhong et al., 2024; Patel et al., 2024), an extra prefill delays only its own request; under colocation, it consumes time that would otherwise serve decoding (Appendix Table 3 models both). Because the gain disappears once prefill competes for capacity, the multi-prefill policy targets disaggregated serving, an architecture increasingly adopted for large deployments.

5 Conclusion

Ultimately, serving a quantized MoE model means choosing a point on the trade-off between quality and throughput, and a static deployment commits every request to the same point. Our framework instead moves along this trade-off per request, and it does so in a principled way at three levels. At the loss level, quality loss factors into request-independent expert fragility and request-dependent expert affinity. At the signal level, FWP reads this loss from one prefill pass on the cheapest instance, bringing fixed-budget allocation within 1.5% of routing on the true per-request loss. At the decision level, the LP-derived router routes each request by a score KKT-consistent with the LP optimum. In live serving, it reaches up to $1.20\times$ the throughput of the only static deployment meeting every class budget, and its throughput rises monotonically as the budget loosens, so each class's quality budget directly sets where the service sits on the trade-off.

AI USE STATEMENT

Generative AI tools assisted with software coding, limited manuscript language polishing, ICLR formatting, and a preliminary proof-and-claim consistency audit. The research ideas and hypotheses were developed by the authors. The authors take responsibility for the final content, including AI-assisted text and code.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 131000–131034, 2024. doi: 10.52202/079017-4164.

Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 117–134, 2024.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021. doi: 10.48550/arXiv.2107.07511.

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models. *Mobius Labs Technical Report*, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021. doi: 10.1145/3478535.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

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 *International Conference on Learning Representations*, 2026.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1280–1297, 2024. doi: 10.18653/v1/2024.acl-long.70.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. doi: 10.48550/arXiv.2412.19437.

DeepSeek-AI et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. doi: 10.48550/arXiv.2405.04434.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332, 2022. doi: 10.52202/068431-2198.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023. doi: 10.52202/075280-0441.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian AWare quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 293–302, 2019. doi: 10.1109/ICCV.2019.00038.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5547–5569, 2022.

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.

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, 2023.

Elias Frantar and Dan Alistarh. QMoE: Sub-1-bit compression of trillion parameter models. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 6, pp. 439–451, 2024.

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

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4878–4887, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024. doi: 10.48550/arXiv.2403.12031.

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. In *International Conference on Learning Representations*, 2025.

Albert Q. Jiang et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. doi: 10.48550/arXiv.2401.04088.

Young Jin Kim, Raffy Fahim, and Hany Hassan Awadalla. Mixture of quantized experts (MoQE): Complementary effect of low-bit quantization and robustness. *arXiv preprint arXiv:2310.02410*, 2023. doi: 10.48550/arXiv.2310.02410.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

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.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274, 2021.

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

Zhuohan Li, Lianmin Zheng, Yinmin Zhong, Vincent Liu, Ying Sheng, Xin Jin, Yanping Huang, Zhifeng Chen, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. AlpaServe: Statistical multiplexing with model parallelism for deep learning serving. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pp. 663–679, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 6, pp. 87–100, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024. doi: 10.1145/3620666.3651335.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36029–36047, 2024.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025.

OpenRouter. Provider routing. `https://openrouter.ai/docs/guides/routing/provider-selection`, 2026. Documentation. Accessed September 25, 2026.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative LLM inference using phase splitting. In *ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 118–132, 2024. doi: 10.1109/ISCA59077.2024.00019.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. doi: 10.48550/arXiv.2407.10671.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. doi: 10.48550/arXiv.2404.02258.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17555–17566, 2021.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472, 2022. doi: 10.52202/068431-1269.

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

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.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. FlexGen: High-throughput generative inference of large language models with a single GPU. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31094–31116, 2023.

Biao Sun, Ziming Huang, Hanyu Zhao, Wencong Xiao, Xinyi Zhang, Yong Li, and Wei Lin. Llumnix: Dynamic scheduling for large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 173–191, 2024.

vLLM Team. Data parallel deployment. `https://docs.vllm.ai/en/stable/serving/data_parallel_deployment/`, 2026. Documentation. Accessed September 25, 2026.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8604–8612, 2019. doi: 10.1109/CVPR.2019.00881.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, 2023.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27168–27183, 2022. doi: 10.52202/068431-1970.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for transformer-based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, 2022.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 62557–62583, 2024. doi: 10.52202/079017-2000.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 193–210, 2024.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022. doi: 10.52202/068431-0515.

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. doi: 10.48550/arXiv.2202.08906.

A Extended Related Work and Positioning

This appendix expands the compact discussion in Section 2.2 along the four lines of work that frame our contribution.

Sparse MoE architectures and routing. The conditional-computation idea behind sparsely activated experts (Shazeer et al., 2017) was scaled and stabilized through GShard, Switch Transformers, GLaM, and ST-MoE (Lepikhin et al., 2021; Fedus et al., 2022; Du et al., 2022; Zoph et al., 2022), with router-design alternatives such as BASE Layers, Hash Layers, and Expert-Choice routing addressing load balance and stability (Lewis et al., 2021; Roller et al., 2021; Zhou et al., 2022). Open competitive MoE checkpoints such as Mixtral, DeepSeekMoE, DeepSeek-V2/V3, and Qwen MoE now place expert specialization at the center of LLM design (Jiang et al., 2024; Dai et al., 2024; DeepSeek-AI et al., 2024; DeepSeek-AI, 2024; Qwen Team, 2024; 2025), and Mixture-of-Depths extends the same routing logic to token-level compute allocation (Raposo et al., 2024). We treat this body of work as the architectural substrate. Our contribution is not a new MoE architecture or training recipe; we ask how heterogeneously quantized instances of a given MoE checkpoint should be evaluated and routed at serving time, once expert affinity, expert fragility, and request heterogeneity are measured.

Post-training quantization and expert-wise bit allocation for LLMs. Low-bit LLM compression has matured rapidly through methods including LLM.int8(), ZeroQuant, GPTQ, SmoothQuant, AWQ, SparseGPT, OmniQuant, QLoRA, SpinQuant, and HQQ (Dettmers et al., 2022; Yao et al., 2022; Frantar et al., 2023; Xiao et al., 2023; Lin et al., 2024; Frantar & Alistarh, 2023; Shao et al., 2024; Dettmers et al., 2023; Liu et al., 2025; Badri & Shaji, 2023). Non-uniform bit allocation has long been informed by Hessian- and sensitivity-based methods such as HAWQ, HAQ, and BRECQ (Dong et al., 2019; Wang et al., 2019; Li et al., 2021), which provide the conceptual ancestry for the per-expert fragility surrogate we evaluate. MoE-specific quantization is now its own subline: QMoE compresses trillion-parameter sparse models to sub-bit precision, MoQE and MC-MoE study expert-wise bit assignment, and Chowdhury et al. (2026) prove that an expert's change in router-vector ℓ_2 norm during training, Λ_s, is a theoretically grounded indicator of which experts must be kept at higher bitwidth (Frantar & Alistarh, 2024; Kim et al., 2023; Huang et al., 2025; Chowdhury et al., 2026). The FWP proxy in Section 3 uses final router norms as measured proxies for expert fragility and lifts this offline per-expert prior to an online per-request quality score. Our paper is complementary: rather than proposing another quantizer, we assume that a small pool of quantized instances has already been materialized by these methods and study the downstream decision problem of which instance should serve which request and which quality signal is reliable enough to drive that decision. A checkpoint-level quality average can be an excellent audit number while still being too coarse for workload-conditioned routing, which is the gap our FWP signal targets.

LLM serving systems and request scheduling. Production-grade serving systems have made LLM inference dramatically more efficient through iteration-level scheduling, paged attention, offloading, stall-free batching, prefill/decode disaggregation, autoscaling, and program-level runtime optimization, including Orca, vLLM, FlexGen, Sarathi-Serve, DistServe, SGLang, AlpaServe, Splitwise, and Llumnix (Yu et al., 2022; Kwon et al., 2023; Sheng et al., 2023; Agrawal et al., 2024; Zhong et al., 2024; Zheng et al., 2024; Li et al., 2023; Patel et al., 2024; Sun et al., 2024), with speculative-decoding methods further reshaping the latency/throughput frontier (Leviathan et al., 2023; Miao et al., 2024). OpenRouter supports provider selection with quantization filters, while vLLM exposes MoE data-parallel endpoints for external request routing (OpenRouter, 2026; vLLM Team, 2026). We rely on the systems insight that throughput and latency are workload properties, not static model properties. The missing axis in most serving evaluations is quantization-induced quality risk; our routing experiments add a request-level quality term to the usual latency and queueing terms while remaining careful not to claim a full production scheduler.

Adaptive inference and request-level model routing. A growing line of work routes requests across heterogeneous models to trade quality for cost: FrugalGPT cascades models by confidence (Chen et al., 2024), RouteLLM and RouterBench learn binary or multi-way model selectors (Ong et al., 2025; Hu et al., 2024), Hybrid-LLM and AutoMix mix small and large models on a per-request basis (Ding et al., 2024; Aggarwal et al., 2024), and CALM uses early-exit confidence inside a single model (Schuster et al., 2022). Closest in spirit, these routers select *which model* to call; we route

across *which bitwidth instance of the same MoE checkpoint*, and the decision lever is the expert-fragility lens rather than aggregate accuracy gaps between distinct models. Their confidence-based scoring also assumes that the candidate models are themselves well-calibrated outputs, whereas in our setting the very object being predicted (per-request quality degradation under a quantized instance) is what the FWP proxy is designed to measure.

Calibration, selective prediction, and risk-controlled inference. Our use of FWP proxies sits within a longer line on calibration and selective prediction, which insists that a model's confidence or risk score be evaluated against downstream decisions rather than only against aggregate accuracy (Guo et al., 2017; Geifman & El-Yaniv, 2017). Distribution-free risk control via conformal prediction and Risk-Controlling Prediction Sets makes those guarantees finite-sample (Angelopoulos & Bates, 2021; Bates et al., 2021), and recent work extends conformal control to language-model outputs and generation-time abstention (Quach et al., 2024; Mohri & Hashimoto, 2024). In our setting the calibrated object is request-level quality degradation under a quantized MoE instance, not class probability; the small audited quality-risk budget in our routing policy is the operational analog of a target risk level. Static average NLL and perplexity remain necessary audit targets, but the decisive question is whether a proxy helps a serving policy reduce workload-conditioned utility gap under realistic information and cost assumptions.

B A Self-Contained Derivation of the Reduced-Form Quality Surrogate Under Explicit Model Assumptions

This appendix replaces the generic perturbation argument with a fully specified two-expert sparse-MoE model. The development is staged in the standard order: first the data-generating process, router, expert margins, and quantization noise; then the induced quality-loss expression; and finally the connection from the resulting coefficient map to the structured proxy used in the main paper.

B.1 Proof Architecture

The derivation has five steps.

1. Specify a two-expert sparse MoE in which requests belong to a rare population of mass α and a common population of mass $1-\alpha$, and each population is routed to its own expert.
2. Define quality as negative expected logistic loss and show that the quality loss under quantization decomposes exactly into an affinity term and an expert-local degradation term.
3. Apply a small-noise expansion to each local degradation term to obtain a reduced-form surrogate of the form "coefficient $\times$ bit-decay", then connect the coefficient to a router-derived margin proxy.
4. Instantiate the same model with FWP-style weights and prove that the weighted proxy amplifies the degradation of the rare, fragile expert.
5. Convert the same weighted surrogate into a request-level utility comparison, which makes the connection to the routing objective in Eq. (103) explicit.

The gain of this route is that the additive surrogate is no longer postulated first and justified later. It is derived from an explicit mathematical model.

The appendix is local rather than global. It justifies the form of the empirical surrogate and the FWP-style amplification mechanism under transparent assumptions; it does not prove that real MoE checkpoints obey an exact additive law globally. The main text accordingly treats the fitted surrogate as a weak ranking prior and uses measured FWP signals for the strongest routing claims.

B.2 Two-Expert Sparse-MoE Model

We consider a sparse top-1 MoE classifier with two experts, denoted by e_r and e_c. Each request belongs to one of two populations:

$$X \in \{r, c\}, \qquad \Pr(X = r) = \alpha, \qquad \Pr(X = c) = 1 - \alpha, \qquad 0 < \alpha < \frac{1}{4}. \tag{14}$$

Population r is the rare population and population c is the common population. The router is deterministic in this model:

$$g(x) = \begin{cases} e_r, & X = r, \\ e_c, & X = c. \end{cases} \tag{15}$$

This is the simplest mathematical extraction of sparse MoE behavior: each request only activates a small subset of experts, and in the present specification that subset has size one.

For a request of population $j \in \{r, c\}$, let $m_j > 0$ be the full-precision signed classification margin of the routed expert. We assume

$$0 < m_r < m_c, \tag{16}$$

which captures the fragility mechanism of Chowdhury et al. (2026) in its simplest form: the rare expert has a weaker activation margin and is therefore more fragile under quantization perturbations.

The prediction loss is logistic:

$$\ell(z) := \log\bigl(1 + e^{-z}\bigr). \tag{17}$$

The full-precision quality is defined by

$$Q_{\text{full}} := -\mathbb{E}\bigl[\ell(m_X)\bigr]. \tag{18}$$

B.3 Quantization Model

If expert e_j is stored at bitwidth b_j, quantization perturbs the signed margin by a scalar random variable $\xi_j(b_j)$, so that the quantized margin for a request of type j becomes

$$M_j(b_j) := m_j + \xi_j(b_j). \tag{19}$$

We assume the following noise model for each $j \in \{r, c\}$:

$$\mathbb{E}[\xi_j(b_j)] = 0, \tag{20}$$

$$\mathbb{E}[\xi_j(b_j)^2] = v_j 2^{-2b_j}, \tag{21}$$

$$\mathbb{E}[|\xi_j(b_j)|^3] \le \chi_j 2^{-3b_j}, \tag{22}$$

$$|\xi_j(b_j)| \le \bar{\delta}_j 2^{-b_j} \quad \text{a.s.} \tag{23}$$

for constants $v_j, \chi_j, \bar{\delta}_j > 0$. The variance scaling in Eq. (23) is the standard uniform-quantization scaling under a mean-square criterion.

Define the quantized quality by

$$Q(\mathbf{b}) := -\mathbb{E}\bigl[\ell(M_X(b_X))\bigr], \qquad \mathbf{b} = (b_r, b_c). \tag{24}$$

The induced quality loss is

$$\Delta Q(\mathbf{b}) := Q_{\text{full}} - Q(\mathbf{b}). \tag{25}$$

B.4 Exact Affinity and Fragility Decomposition

The first key fact is exact rather than asymptotic.

Proposition B.1 (Exact mixture decomposition)**.** Under the model above,

$$\Delta Q(\mathbf{b}) = \alpha\, \Delta_r(b_r) + (1 - \alpha)\, \Delta_c(b_c), \tag{26}$$

where

$$\Delta_j(b_j) := \mathbb{E}\bigl[\ell(M_j(b_j)) - \ell(m_j)\bigr], \qquad j \in \{r, c\}. \tag{27}$$

Proof. Condition on the request type X. By definition,

$$\Delta Q(\mathbf{b}) = \mathbb{E}\Big[\ell(M_X(b_X)) - \ell(m_X)\Big]. \tag{28}$$

Using the law of total expectation and the probabilities of the two populations gives

$$\Delta Q(\mathbf{b}) = \Pr(X = r)\, \mathbb{E}\bigl[\ell(M_r(b_r)) - \ell(m_r)\bigr] + \Pr(X = c)\, \mathbb{E}\bigl[\ell(M_c(b_c)) - \ell(m_c)\bigr], \tag{29}$$

which is exactly Eq. (26). □

Proposition B.1 already factors into two objects that should not be combined. The probabilities α and $1 - \alpha$ are affinity weights. The local terms Δ_r and Δ_c are intrinsic degradation functions of the routed experts.

B.5 Small-Noise Expansion of the Local Degradation

We now derive the reduced form by expanding the local term in Eq. (27). Let

$$\underline{m}_j := m_j - \bar{\delta}_j 2^{-b_j}. \tag{30}$$

Assume $\underline{m}_j > 0$, so the quantized margin remains in a compact subset of the positive half-line. Define

$$L_j := \sup_{z \geq \underline{m}_j} |\ell'''(z)|. \tag{31}$$

Lemma B.1 (Second-order local expansion)**.** For each $j \in \{r, c\}$,

$$\Delta_j(b_j) = \frac{1}{2}\ell''(m_j)v_j 2^{-2b_j} + r_j(b_j), \tag{32}$$

where the remainder satisfies

$$|r_j(b_j)| \leq \frac{L_j}{6}\chi_j 2^{-3b_j}. \tag{33}$$

Proof. Taylor-expand $\ell(m_j + \xi_j)$ around m_j:

$$\ell(m_j + \xi_j) = \ell(m_j) + \ell'(m_j)\xi_j + \frac{1}{2}\ell''(m_j)\xi_j^2 + \frac{1}{6}\ell'''(\widetilde{m}_j)\xi_j^3, \tag{34}$$

where $\widetilde{m}_j$ lies between m_j and $m_j + \xi_j$. Taking expectations and using $\mathbb{E}[\xi_j] = 0$ removes the first-order term. The second-order term becomes $\frac{1}{2}\ell''(m_j)\mathbb{E}[\xi_j^2] = \frac{1}{2}\ell''(m_j)v_j 2^{-2b_j}$. For the remainder,

$$\left|\frac{1}{6}\mathbb{E}\big[\ell'''(\widetilde{m}_j)\xi_j^3\big]\right| \leq \frac{L_j}{6}\mathbb{E}[|\xi_j|^3] \leq \frac{L_j}{6}\chi_j 2^{-3b_j}, \tag{35}$$

which proves Eq. (33). □

Lemma B.1 is the precise point where bitwidth enters the theory. The leading term is proportional to 2^{-2b_j} because the dominant contribution is the variance of the quantization noise. If one works instead with a worst-case absolute perturbation criterion, the same derivation produces a 2^{-b_j} decay. This is why the main text keeps the bit-response map as a generic function $\phi(b)$.

Lemma B.2 (Biased-noise first-order variant)**.** Suppose instead that the quantization perturbation has a small mean-shift component

$$\mathbb{E}[\xi_j(b_j)] = m_j^{(1)} 2^{-b_j} + m_j^{(2)} 2^{-2b_j} + \mathcal{O}(2^{-3b_j}), \tag{36}$$

while its second and third moment bounds follow the scaling in Eq. (23). Then the local degradation admits the expansion

$$\Delta_j(b_j) = \ell'(m_j)m_j^{(1)} 2^{-b_j} + \left(\ell'(m_j)m_j^{(2)} + \frac{1}{2}\ell''(m_j)v_j\right) 2^{-2b_j} + \mathcal{O}(2^{-3b_j}). \tag{37}$$

Proof. Use the same Taylor expansion as in Lemma B.1, but do not cancel the first-order term. Taking expectations gives

$$\Delta_j(b_j) = \ell'(m_j)\mathbb{E}[\xi_j(b_j)] + \frac{1}{2}\ell''(m_j)\mathbb{E}[\xi_j(b_j)^2] + \mathcal{O}(\mathbb{E}|\xi_j(b_j)|^3). \tag{38}$$

Substituting the assumed mean expansion and the moment bounds yields Eq. (37). □

Lemma B.2 explains why the experiments compare both 2^{-b} and 2^{-2b} families. If the dominant distortion behaves like symmetric zero-mean noise, the energy-response term 2^{-2b} is natural. If clipping, scaling, or quantizer bias introduces a first-order mean shift, a 2^{-b} term can dominate over the measured range. The empirical comparison between the two families is therefore part of the model check rather than an arbitrary hyperparameter sweep.

Theorem B.1 (Reduced form under explicit model assumptions)**.** Under Proposition B.1 and Lemma B.1,

$$\Delta Q(\mathbf{b}) = \alpha s_r 2^{-2b_r} + (1-\alpha) s_c 2^{-2b_c} + r(\mathbf{b}), \tag{39}$$

where $s_j := \frac{1}{2} v_j \ell''(m_j)$ for $j \in \{r, c\}$, and

$$|r(\mathbf{b})| \le \frac{L_r}{6} \alpha \chi_r 2^{-3b_r} + \frac{L_c}{6} (1-\alpha) \chi_c 2^{-3b_c}. \tag{40}$$

Proof. Substitute the two local expansions from Eq. (32) into the exact decomposition in Eq. (26) and collect terms. □

Theorem B.1 is the core constructive result. The reduced-form surrogate is not assumed. It is obtained by combining an exact affinity decomposition with a local Taylor expansion under symmetric quantization noise.

B.6 Intrinsic Fragility and Rare-Expert Ordering

To factor intrinsic fragility out of the affinity-weighted coefficient, recall the intrinsic local coefficient

$$s_j := \frac{1}{2} v_j \ell''(m_j), \qquad j \in \{r, c\}. \tag{41}$$

The leading coefficients in Eq. (39) are therefore αs_r and $(1-\alpha)s_c$, explicitly factoring the population-level affinity from the intrinsic fragility.

Proposition B.2 (Rare expert is intrinsically more fragile)**.** If $0 < m_r < m_c$ and $v_r \ge v_c$, then

$$s_r > s_c. \tag{42}$$

Proof. For $z > 0$, the logistic loss satisfies

$$\ell''(z) = \frac{e^z}{(1+e^z)^2}, \qquad \frac{d}{dz}\ell''(z) = \frac{e^z(1-e^z)}{(1+e^z)^3} < 0. \tag{43}$$

Hence $\ell''(m_r) > \ell''(m_c)$ whenever $0 < m_r < m_c$. Multiplying by $v_r/2$ and $v_c/2$ and using $v_r \ge v_c$ yields $s_r > s_c$. □

Proposition B.2 formalizes the qualitative statement that weaker-margin experts are more fragile under the same quantization-noise scale. Affinity determines how often that fragility matters; it does not define fragility itself.

B.7 Connection to Router-Based Structured Coefficients

One bridge back to the main paper remains: how should the margin m_j be represented by observable router-side quantities? The next assumption makes that step explicit rather than implicit.

Assumption B.1 (Router-margin link)**.** For each population-specific expert $j \in \{r, c\}$, there exist constants $\kappa, \mu, \tau > 0$ and a scale correction variable $V_j \ge 0$ such that

$$m_j \ge \kappa \frac{\widetilde{\Lambda}_j + \tau}{1 + \mu V_j}, \tag{44}$$

where $\widetilde{\Lambda}_j$ is a router-derived margin proxy. This abstracts the claim in Chowdhury et al. (2026) that larger router-norm change corresponds to a larger activation margin.

Proposition B.3 (Monotone upper bound for structured coefficients)**.** Under Assumption B.1,

$$s_j \le \frac{v_j}{2} \ell'' \left(\kappa \frac{\widetilde{\Lambda}_j + \tau}{1 + \mu V_j} \right), \qquad j \in \{r, c\}. \tag{45}$$

Therefore s_j is upper-bounded by a monotone decreasing function of $(\widetilde{\Lambda}_j + \tau)/(1 + \mu V_j)$.

Proof. Equation (41) gives $s_j = \frac{1}{2} v_j \ell''(m_j)$. Since $\ell''(z)$ is decreasing on $z > 0$, substituting the lower bound on m_j from Eq. (44) yields Eq. (45). The right-hand side is monotone decreasing in $(\widetilde{\Lambda}_j + \tau)/(1 + \mu V_j)$, giving the stated upper-bound interpretation. □

Proposition B.3 is the right interpretation of the structured coefficient family used as a restricted ablation in the main text (Section 3.1). The empirical structured family is

$$\widehat{c}_e^{\,\mathrm{str}} = \theta_{\ell(e)} \left(\frac{1 + \mu V_e}{\widetilde{\Lambda}_e + \tau} \right)^p, \tag{46}$$

where $\widetilde{\Lambda}_e$ is a router-derived fragility proxy (final router norm), V_e is a normalized-MaxVar scale correction, and (μ, τ, p) are shared parameters. The model does not prove that the exact coefficient must equal a specific power law of $(1 + \mu V_j)/(\widetilde{\Lambda}_j + \tau)$. It proves a cleaner statement: the intrinsic fragility coefficient is controlled by a monotone decreasing upper bound in a router-derived margin proxy. The parametric family in Eq. (46) is then a parsimonious approximation to that monotone upper-bound shape when we return to real MoE checkpoints with many experts and imperfect observability.

B.8 FWP-Style Weighting and Fragile-Expert Amplification

We next instantiate the same model with the FWP-style weighting scheme from Eq. (6). Since each request is routed to exactly one expert, the token-level weights collapse to population-level constants. Define

$$\omega_r = \frac{(\widetilde{\Lambda}_r + \epsilon_w)^{-1}}{\alpha(\widetilde{\Lambda}_r + \epsilon_w)^{-1} + (1-\alpha)(\widetilde{\Lambda}_c + \epsilon_w)^{-1}}, \qquad \omega_c = \frac{(\widetilde{\Lambda}_c + \epsilon_w)^{-1}}{\alpha(\widetilde{\Lambda}_r + \epsilon_w)^{-1} + (1-\alpha)(\widetilde{\Lambda}_c + \epsilon_w)^{-1}}. \tag{47}$$

Then

$$\alpha\omega_r + (1-\alpha)\omega_c = 1, \tag{48}$$

and if $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$, then $\omega_r > 1 > \omega_c$.

The two-expert analogue of the log-FWP increment in Eq. (8) is

$$\Delta\ell_{\mathrm{FWP}}^{\mathrm{model}}(\mathbf{b}) := \alpha\omega_r \Delta_r(b_r) + (1-\alpha)\omega_c \Delta_c(b_c). \tag{49}$$

The ordinary average NLL increment is exactly the quality loss from Eq. (25), namely $\Delta Q(\mathbf{b}) = \alpha\Delta_r(b_r) + (1-\alpha)\Delta_c(b_c)$.

Proposition B.4 (FWP amplifies fragile-expert degradation)**.** Suppose $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$ and $\Delta_r(b_r) \geq \Delta_c(b_c)$. Then

$$\Delta\ell_{\mathrm{FWP}}^{\mathrm{model}}(\mathbf{b}) - \Delta Q(\mathbf{b}) = \alpha(\omega_r - 1)\big(\Delta_r(b_r) - \Delta_c(b_c)\big) \geq 0. \tag{50}$$

If both inequalities are strict, then the inequality in Eq. (50) is strict.

Proof. Subtract Eq. (26) from Eq. (49):

$$\Delta\ell_{\mathrm{FWP}}^{\mathrm{model}}(\mathbf{b}) - \Delta Q(\mathbf{b}) = \alpha(\omega_r - 1)\Delta_r(b_r) + (1-\alpha)(\omega_c - 1)\Delta_c(b_c). \tag{51}$$

Using the normalization identity in Eq. (48), we have $(1-\alpha)(\omega_c - 1) = -\alpha(\omega_r - 1)$, so the difference reduces to Eq. (50). Since $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$ implies $\omega_r > 1$, the sign is determined by $\Delta_r(b_r) - \Delta_c(b_c)$. □

Proposition B.4 is the structural counterpart of the empirical claim behind FWP: if requests routed to the rare expert already suffer larger degradation, inverse-router weighting makes that fragile-expert degradation more visible rather than washing it out.

Theorem B.2 (FWP-weighted reduced form)**.** Under Lemma B.1, the FWP-style proxy increment admits the reduced form

$$\Delta\ell_{\mathrm{FWP}}^{\mathrm{model}}(\mathbf{b}) = \widetilde{c}_r 2^{-2b_r} + \widetilde{c}_c 2^{-2b_c} + \widetilde{r}(\mathbf{b}), \tag{52}$$

where

$$\widetilde{c}_r := \omega_r \alpha s_r, \qquad \widetilde{c}_c := \omega_c (1-\alpha) s_c, \tag{53}$$

and

$$\widetilde{r}(\mathbf{b}) := \alpha\omega_r r_r(b_r) + (1-\alpha)\omega_c r_c(b_c). \tag{54}$$

Moreover,

$$\frac{\widetilde{c}_r}{\widetilde{c}_c} = \frac{\omega_r}{\omega_c}\frac{\alpha s_r}{(1-\alpha)s_c} > \frac{\alpha s_r}{(1-\alpha)s_c} \qquad \text{whenever } \widetilde{\Lambda}_r < \widetilde{\Lambda}_c. \tag{55}$$

Proof. Insert the local expansion from Eq. (32) into Eq. (49). The coefficient formula follows by collecting the 2^{-2b_j} terms. Since $\omega_r > \omega_c > 0$ when $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$, Eq. (55) follows immediately. □

Theorem B.2 links the first appendix back to the main text in two places. First, it reproduces the metric-agnostic reduced form from Eq. (4) under the concrete FWP instantiation in Eq. (8). Second, it shows why the same router-derived variables can matter more strongly under a weighted proxy than under plain average NLL: the rare-expert coefficient is inflated by the factor ω_r/ω_c.

B.9 Two-Expert Implication for Request-Level Utility

The same model also connects directly to the policy layer in Section 3.3. Consider two pre-quantized instances, H and L, with latencies $T_H > T_L$. Let their bitwidths satisfy $b_j^{(H)} > b_j^{(L)}$ for each population-specific expert $j \in \{r, c\}$. For either proxy $\mathsf{proxy} \in \{\mathrm{avg}, \mathrm{FWP}\}$, define the one-request utility

$$U_j^{\mathsf{proxy}}(a) = R_j^{\mathrm{base}} - \alpha_q\, \Delta\ell_j^{\mathsf{proxy}}(a) - \beta T_a, \tag{56}$$

where

$$\Delta\ell_j^{\mathrm{avg}}(a) := \Delta_j\big(b_j^{(a)}\big), \qquad \Delta\ell_j^{\mathrm{FWP}}(a) := \omega_j \Delta_j\big(b_j^{(a)}\big). \tag{57}$$

The utility advantage of routing type-j requests to the higher-bitwidth instance is

$$\Gamma_j^{\mathsf{proxy}} := U_j^{\mathsf{proxy}}(H) - U_j^{\mathsf{proxy}}(L). \tag{58}$$

Proposition B.5 (FWP enlarges routing separation under the explicit model)**.** Let

$$\delta_j := 2^{-2b_j^{(L)}} - 2^{-2b_j^{(H)}} > 0. \tag{59}$$

Then, under Lemma B.1,

$$\Gamma_j^{\mathrm{avg}} = \alpha_q s_j \delta_j - \beta(T_H - T_L) + \zeta_j^{\mathrm{avg}}, \tag{60}$$

$$\Gamma_j^{\mathrm{FWP}} = \alpha_q \omega_j s_j \delta_j - \beta(T_H - T_L) + \zeta_j^{\mathrm{FWP}}, \tag{61}$$

where

$$|\zeta_j^{\mathrm{avg}}| \le \alpha_q \frac{L_j}{6}\chi_j\big(2^{-3b_j^{(L)}} + 2^{-3b_j^{(H)}}\big), \tag{62}$$

$$|\zeta_j^{\mathrm{FWP}}| \le \alpha_q \omega_j \frac{L_j}{6}\chi_j\big(2^{-3b_j^{(L)}} + 2^{-3b_j^{(H)}}\big). \tag{63}$$

If $\widetilde{\Lambda}_r < \widetilde{\Lambda}_c$ and $s_r > s_c$, then for sufficiently large bitwidths the utility separation between rare and common requests is strictly larger under FWP than under average NLL:

$$\Gamma_r^{\mathrm{FWP}} - \Gamma_c^{\mathrm{FWP}} > \Gamma_r^{\mathrm{avg}} - \Gamma_c^{\mathrm{avg}}. \tag{64}$$

Proof. The identities in Eq. (61) follow by inserting the local expansions from Lemma B.1 into Eq. (58). For the separation claim, subtract the two proxy gaps:

$$\big(\Gamma_r^{\mathrm{FWP}} - \Gamma_c^{\mathrm{FWP}}\big) - \big(\Gamma_r^{\mathrm{avg}} - \Gamma_c^{\mathrm{avg}}\big) = \alpha_q\Big[(\omega_r - 1)s_r\delta_r + (1-\omega_c)s_c\delta_c\Big] + \big(\zeta_r^{\mathrm{FWP}} - \zeta_c^{\mathrm{FWP}} - \zeta_r^{\mathrm{avg}} + \zeta_c^{\mathrm{avg}}\big). \tag{65}$$

The leading term is strictly positive because $\omega_r > 1$, $\omega_c < 1$, $s_r > 0$, and $s_c > 0$. For sufficiently large bitwidths, the remainder term is dominated by the leading term; equivalently, the claim holds once the absolute remainder combination is smaller than the positive leading separation. This gives Eq. (64). □

Proposition B.5 is the direct bridge to the later policy appendix. Under the explicit model, FWP does not merely increase the measured degradation of rare requests. It widens the utility gap between routing rare requests to a higher-bitwidth instance and routing common requests to that higher-bitwidth instance. This is exactly the kind of separation that a downstream routing policy can exploit.

C Derivation of a Window-Level LP Routing Template

This appendix gives the operations-research derivation behind the LP routing template in Section 3.3. The goal is not only to explain where Eq. (12) comes from, but also to make the full technical chain explicit: primal LP, exact dual program, shadow-price characterization, subgradient price dynamics, proximal quota regularization, and calibration-to-utility transfer. The routing experiments instantiate the same utility ingredients in greedy simulators; they do not claim to implement this full window-level scheduler.

C.1 Primal and Dual Linear Programs

Consider the LP in Eq. (10). For convenience, write the quality-constraint coefficient as

$$q_{ga}(t) := \widehat{\ell}_{ga}(t) - \tau_g. \tag{66}$$

Proposition C.1 (Dual LP and strong duality)**.** For a fixed control window t, the dual of the primal LP in Eq. (10) is

$$\begin{aligned} \min_{\lambda,\pi,\nu\geq 0} \quad & \sum_{g\in\mathcal{G}} \lambda_g B_g(t) + \sum_{a\in\mathcal{A}} \pi_a C_a(t) \\ \text{s.t.} \quad & \lambda_g + \pi_a s_{ga}(t) + \nu_g q_{ga}(t) \geq L_g, \qquad \forall g\in\mathcal{G},\ a\in\mathcal{A}. \end{aligned} \tag{67}$$

Whenever Eq. (10) is feasible and bounded, strong duality holds.

Proof. Write Eq. (10) in the standard form

$$\max_{x\geq 0}\ c^\top x \qquad \text{s.t.} \quad Ax \leq b, \tag{68}$$

where x stacks the variables $x_{ga}(t)$, the vector c collects the throughput coefficients L_g, and A collects the demand, capacity, and quality-constraint coefficients. The stated dual is then the standard LP dual with nonnegative multipliers for each inequality block. Strong duality follows from the fundamental theorem of linear programming. □

Proposition C.1 is the OR foundation of the online rule. The prices λ_g, π_a, and ν_g are not free tuning terms; they are the exact dual multipliers of demand, capacity, and quality constraints.

Proposition C.2 (Shadow-price characterization of active routes)**.** Let x^* solve the primal LP in Eq. (10), and let $(\lambda^*, \pi^*, \nu^*)$ solve the dual LP in Eq. (67). Then for every class-instance pair (g, a),

$$x^*_{ga}(t) > 0 \quad \Longrightarrow \quad L_g - \pi^*_a s_{ga}(t) - \nu^*_g q_{ga}(t) = \lambda^*_g. \tag{69}$$

If $x^*_{ga}(t) = 0$, then

$$L_g - \pi^*_a s_{ga}(t) - \nu^*_g q_{ga}(t) \leq \lambda^*_g. \tag{70}$$

Consequently, a class may be split only across instances that maximize the reduced reward at the optimal shadow prices.

Proof. The complementary-slackness condition for the nonnegativity constraint on $x_{ga}(t)$ implies

$$x^*_{ga}(t)\Big[\lambda^*_g + \pi^*_a s_{ga}(t) + \nu^*_g q_{ga}(t) - L_g\Big] = 0. \tag{71}$$

If $x^*_{ga}(t) > 0$, the bracketed term must vanish, which yields Eq. (69). If $x^*_{ga}(t) = 0$, dual feasibility gives Eq. (70). □

Proposition C.2 is the class-level shadow-price characterization. It is the exact static analogue of the online rule: active routes are precisely those instances whose shadow-price-adjusted utility is maximal.

C.2 Lagrangian Decomposition and One-Step Reduced Reward

The corresponding Lagrangian is

$$\mathcal{L}(x,\lambda,\pi,\nu)=\sum_{g,a}\Big[L_g-\lambda_g-\pi_a s_{ga}(t)-\nu_g q_{ga}(t)\Big]x_{ga}(t)+\sum_g \lambda_g B_g(t)+\sum_a \pi_a C_a(t). \quad (72)$$

Proposition C.3 (Reduced reward decomposition)**.** Fix nonnegative dual variables (λ,π,ν). For any request class g, maximizing the Lagrangian contribution of a single incremental assignment is equivalent to choosing the instance with largest reduced reward

$$R_{ga}(t)=L_g-\pi_a(t)s_{ga}(t)-\nu_g(t)q_{ga}(t), \quad (73)$$

up to the class-specific constant λ_g.

Proof of Proposition C.3. Fix nonnegative dual variables (λ,π,ν). Consider a single incremental assignment of one unit of class-g traffic. Let $\delta_a\in\{0,1\}$ indicate whether that unit is sent to instance a, with $\sum_a \delta_a=1$. The Lagrangian contribution of this one-step assignment is

$$\sum_{a\in\mathcal{A}}\Big[L_g-\lambda_g-\pi_a s_{ga}(t)-\nu_g q_{ga}(t)\Big]\delta_a. \quad (74)$$

Since the class-specific term λ_g does not depend on a, maximizing over the simplex is equivalent to choosing the instance with largest coefficient

$$R_{ga}(t)=L_g-\pi_a(t)s_{ga}(t)-\nu_g(t)q_{ga}(t). \quad (75)$$

This is exactly the reduced reward stated in Proposition C.3. □

Proposition C.3 is the one-step version of Proposition C.2. The multiplier λ_g acts as a class-specific reservation value and drops out when comparing instances within the same class, which is why the online score only needs the capacity and quality prices.

C.3 Soft Quality Enforcement and the Hinge Penalty

A common variant replaces the signed coefficient $q_{ga}(t)$ with the one-sided penalty $\max\{0,\widehat{\ell}_{ga}(t)-\tau_g\}$, the standard slack-variable representation of a soft quality constraint.

For a one-step assignment, introduce a slack variable $\xi_{ga}\geq 0$ satisfying

$$\xi_{ga}\geq \widehat{\ell}_{ga}(t)-\tau_g. \quad (76)$$

Penalizing this slack at rate ν_g yields the local objective term

$$-\nu_g\xi_{ga}. \quad (77)$$

Eliminating ξ_{ga} by pointwise minimization gives the closed form

$$\xi^\star_{ga}=\max\{0,\widehat{\ell}_{ga}(t)-\tau_g\}, \quad (78)$$

and therefore the one-sided variant of Eq. (12). Feasible assignments incur zero slack penalty, while only violating assignments are penalized. The KKT argument below uses the signed coefficient, which is the Lagrangian's own quality term.

C.4 Reduced Dual Objective and Price Updates

The dual derivation can be pushed one step further by eliminating the class-demand multipliers. For fixed (π,ν), the smallest nonnegative λ_g satisfying Eq. (67) is

$$\lambda_g(\pi,\nu)=\Big[\max_{a\in\mathcal{A}}\big\{L_g-\pi_a s_{ga}(t)-\nu_g q_{ga}(t)\big\}\Big]_+. \quad (79)$$

Substituting Eq. (79) into Eq. (67) gives the reduced dual objective

$$\widetilde{D}_t(\pi,\nu)=\sum_{g\in\mathcal{G}}B_g(t)\Big[\max_{a\in\mathcal{A}}\big\{L_g-\pi_a s_{ga}(t)-\nu_g q_{ga}(t)\big\}\Big]_+ +\sum_{a\in\mathcal{A}}\pi_a C_a(t). \quad (80)$$

Equation (80) shows that the dual optimization problem is entirely driven by reduced rewards.

Lemma C.1 (Subgradient characterization of price residuals)**.** Let $x^*(\pi, \nu)$ be any allocation induced by active maximizers in Eq. (80): for each class g, it places mass $B_g(t)$ on instances attaining a positive maximum reduced reward, places zero mass if the positive part is inactive, and may split ties arbitrarily. Then the vectors

$$g_a^\pi(\pi, \nu) := C_a(t) - \sum_{g \in \mathcal{G}} s_{ga}(t) x_{ga}^*(\pi, \nu), \tag{81}$$

$$g_g^\nu(\pi, \nu) := -\sum_{a \in \mathcal{A}} q_{ga}(t) x_{ga}^*(\pi, \nu) \tag{82}$$

are subgradients of the reduced dual objective $\widetilde{D}_t(\pi, \nu)$.

Proof. The reduced dual objective is a positive part of pointwise maxima of affine functions of (π, ν) plus the linear capacity term. By Danskin's theorem, any gradient of an active affine piece is a valid subgradient of $\widetilde{D}_t$. Holding the active allocation $x^*(\pi, \nu)$ fixed, differentiating with respect to π_a gives $C_a(t) - \sum_g s_{ga}(t) x_{ga}^*(\pi, \nu)$, and differentiating with respect to ν_g gives $-\sum_a q_{ga}(t) x_{ga}^*(\pi, \nu)$. □

Lemma C.1 is the formal source of the dual price updates. A projected subgradient descent step on $\widetilde{D}_t$ has the form

$$\pi_a^{(n+1)} = \Big[\pi_a^{(n)} - \eta_n g_a^\pi(\pi^{(n)}, \nu^{(n)})\Big]_+ = \Big[\pi_a^{(n)} + \eta_n \big(\mathrm{UsedCapacity}_a^{(n)} - C_a\big)\Big]_+, \tag{83}$$

$$\nu_g^{(n+1)} = \Big[\nu_g^{(n)} - \eta_n g_g^\nu(\pi^{(n)}, \nu^{(n)})\Big]_+ = \Big[\nu_g^{(n)} + \eta_n \sum_a q_{ga} x_{ga}^{*(n)}\Big]_+, \tag{84}$$

which is the exact optimization-theoretic analogue of the template price update. An online implementation may replace the class-level aggregate $\sum_a q_{ga} x_{ga}^{*(n)}$ by the realized average-loss residual in the window; this is simply a normalized stochastic estimate of the same subgradient component.

Theorem C.1 (Finite-time dual convergence rate)**.** Fix a control window t and assume that $(B_g(t), s_{ga}(t), q_{ga}(t), C_a(t))$ are stationary within that window. Let $\mathcal{K} \subseteq \mathbb{R}_+^{A+|\mathcal{G}|}$ be a compact convex set containing an optimal dual solution (π^*, ν^*), and suppose that every subgradient of $\widetilde{D}_t$ on $\mathcal{K}$ has Euclidean norm at most G. Consider the projected subgradient iterations

$$(\pi^{(n+1)}, \nu^{(n+1)}) = \Pi_{\mathcal{K}}\Big((\pi^{(n)}, \nu^{(n)}) - \eta_n g^{(n)}\Big), \qquad g^{(n)} \in \partial \widetilde{D}_t(\pi^{(n)}, \nu^{(n)}). \tag{85}$$

Then

$$\min_{1 \le n \le T} \Big(\widetilde{D}_t(\pi^{(n)}, \nu^{(n)}) - \widetilde{D}_t(\pi^*, \nu^*)\Big) \le \frac{\|(\pi^{(1)}, \nu^{(1)}) - (\pi^*, \nu^*)\|_2^2 + G^2 \sum_{n=1}^T \eta_n^2}{2 \sum_{n=1}^T \eta_n}. \tag{86}$$

In particular, if $\eta_n = \eta/\sqrt{n}$, then the bound is $\mathcal{O}((1 + \log T)/\sqrt{T})$; with a horizon-tuned constant step size $\eta_n = \eta/\sqrt{T}$, the bound is $\mathcal{O}(T^{-1/2})$.

Proof. Nonexpansiveness of Euclidean projection gives

$$\|(\pi^{(n+1)}, \nu^{(n+1)}) - (\pi^*, \nu^*)\|_2^2 \le \|(\pi^{(n)}, \nu^{(n)}) - (\pi^*, \nu^*)\|_2^2 - 2\eta_n \langle g^{(n)}, (\pi^{(n)}, \nu^{(n)}) - (\pi^*, \nu^*)\rangle + \eta_n^2 \|g^{(n)}\|_2^2. \tag{87}$$

By convexity of $\widetilde{D}_t$,

$$\widetilde{D}_t(\pi^{(n)}, \nu^{(n)}) - \widetilde{D}_t(\pi^*, \nu^*) \le \langle g^{(n)}, (\pi^{(n)}, \nu^{(n)}) - (\pi^*, \nu^*)\rangle. \tag{88}$$

Substituting this bound into the previous inequality, summing from $n = 1$ to T, and using $\|g^{(n)}\|_2 \le G$ yields

$$2 \sum_{n=1}^T \eta_n \Big(\widetilde{D}_t(\pi^{(n)}, \nu^{(n)}) - \widetilde{D}_t(\pi^*, \nu^*)\Big) \le \|(\pi^{(1)}, \nu^{(1)}) - (\pi^*, \nu^*)\|_2^2 + G^2 \sum_{n=1}^T \eta_n^2. \tag{89}$$

Taking the minimum over $n \le T$ proves Eq. (86). □

Theorem C.1 makes the price dynamics fully technical: the online dual variables are not just interpretable as shadow prices; their dual objective gap obeys the standard projected-subgradient rate under stationary window statistics.

C.5 Quota Tracking as a Regularized Primal Step

Let $p^\star_{ga}(t)$ be the target routing share derived from the LP solution and let $\widetilde{p}_{ga}(t)$ be the current realized share inside the same control window. Consider the regularized one-step objective for a request of class g:

$$\max_{\delta \in \Delta_A} \sum_{a \in \mathcal{A}} \Big[L_g - \pi_a s_{ga}(t) - \nu_g q_{ga}(t) \Big] \delta_a - \frac{\beta}{2} \sum_{a \in \mathcal{A}} \Big(\widetilde{p}_{ga}(t) + \delta_a - p^\star_{ga}(t) \Big)^2, \tag{90}$$

where Δ_A is the probability simplex over instances.

Proposition C.4 (Derivation of the final score)**.** Define the quota deficit

$$d_{ga}(t) := p^\star_{ga}(t) - \widetilde{p}_{ga}(t). \tag{91}$$

If the one-step assignment is integral, i.e., $\delta_a \in \{0, 1\}$ and $\sum_a \delta_a = 1$, then maximizing Eq. (90) is equivalent to choosing the instance with largest score

$$S_{ga}(t) = \beta d_{ga}(t) + L_g - \pi_a(t) s_{ga}(t) - \nu_g(t) \big(\widehat{\ell}_{ga}(t) - \tau_g \big). \tag{92}$$

Proof. Fix an action a and set $\delta_a = 1$, $\delta_{a'} = 0$ for $a' \neq a$. Expanding the quadratic term in Eq. (90) gives

$$-\frac{\beta}{2} \sum_{a' \in \mathcal{A}} \Big(\widetilde{p}_{ga'} - p^\star_{ga'} \Big)^2 - \beta \big(\widetilde{p}_{ga} - p^\star_{ga} \big) - \frac{\beta}{2}. \tag{93}$$

The first and third terms are independent of the chosen action and therefore do not affect the maximizer. Using $d_{ga} = p^\star_{ga} - \widetilde{p}_{ga}$, the action-dependent contribution reduces to βd_{ga}. Combining this with the reward, capacity price, and quality price gives the stated score. □

Proposition C.4 is the formal derivation of the final score. The quota term is not ad hoc; it is the first-order effect of a quadratic tracking penalty around the LP target shares.

C.6 Asymptotic Consistency with the LP

We now complete the standard primal-dual interpretation.

Theorem C.2 (KKT consistency of the template policy with the LP optimum)**.** Suppose that within a control window the workload statistics are stationary, the service-cost and loss predictions are exact, and the projected subgradient updates for (π, ν) converge to an optimal dual solution $(\pi^\star, \nu^\star)$ of Eq. (10). Then:

1. if $\beta = 0$, the online rule in Eq. (13) coincides with the greedy primal maximizer of the Lagrangian relaxation;
2. if $\beta > 0$ and the quota deficits satisfy $\frac{1}{T} \sum_{t=1}^{T} \|d_g(t)\|_1 \to 0$ for every class g, then the time-averaged actions generated by Eq. (13) are KKT-consistent with the LP optimum in Eq. (10).

Proof. When $\beta = 0$, Proposition C.3 implies that the online action is exactly the primal maximizer of the one-step Lagrangian contribution under the converged dual prices. Because those prices are optimal, Proposition C.2 implies that every active route must lie in the primal argmax set induced by the shadow prices. This is exactly the KKT stationarity condition for the window-level LP.

When $\beta > 0$, Proposition C.4 shows that the implemented score is the reduced reward plus a regularization perturbation of size at most $\beta \|d_g(t)\|_\infty$. By assumption, the Cesàro average of the deficits converges to zero, so the average perturbation also vanishes. Therefore the time-averaged action sequence has the same limit points as the unregularized primal-dual iterates associated with the optimal dual solution. Combined with Theorem C.1, this implies that the deployed rule inherits the same optimal shadow-price structure asymptotically, hence satisfies primal feasibility, dual feasibility, and complementary slackness in the limit. □

C.7 Calibration Error and Workload-Conditioned Utility Loss

The last link is from quality-prediction error to the workload-conditioned utility in Eq. (103). For this step, consider the linearized one-step utility

$$u_{ga}(t) := L_g - \alpha_q \ell_{ga}(t) - \beta s_{ga}(t), \tag{94}$$

where $\ell_{ga}(t)$ is the true average quality loss of assigning class g to instance a in window t. Let the predicted utility be

$$\widehat{u}_{ga}(t) := L_g - \alpha_q \widehat{\ell}_{ga}(t) - \beta s_{ga}(t). \tag{95}$$

Proposition C.5 (Prediction error implies one-step utility regret)**.** Assume that for a fixed window,

$$\sup_{g\in\mathcal{G},\ a\in\mathcal{A}} |\widehat{\ell}_{ga}(t) - \ell_{ga}(t)| \le \varepsilon. \tag{96}$$

Let

$$a_g^*(t) \in \arg\max_{a\in\mathcal{A}} u_{ga}(t), \qquad \widehat{a}_g(t) \in \arg\max_{a\in\mathcal{A}} \widehat{u}_{ga}(t). \tag{97}$$

Then the one-step utility loss of acting on the predicted surrogate satisfies

$$0 \le u_{ga_g^*(t)}(t) - u_{g\widehat{a}_g(t)}(t) \le 2\alpha_q \varepsilon. \tag{98}$$

Consequently, over any horizon of T requests whose classes are evaluated with the same error bound, the cumulative regret is at most $2\alpha_q \varepsilon T$.

Proof. For any action a,

$$|u_{ga}(t) - \widehat{u}_{ga}(t)| = \alpha_q |\widehat{\ell}_{ga}(t) - \ell_{ga}(t)| \le \alpha_q \varepsilon. \tag{99}$$

Therefore,

$$u_{ga_g^*(t)}(t) - u_{g\widehat{a}_g(t)}(t) = \big[u_{ga_g^*(t)}(t) - \widehat{u}_{ga_g^*(t)}(t)\big] \tag{100}$$

$$+ \big[\widehat{u}_{ga_g^*(t)}(t) - \widehat{u}_{g\widehat{a}_g(t)}(t)\big] \tag{101}$$

$$+ \big[\widehat{u}_{g\widehat{a}_g(t)}(t) - u_{g\widehat{a}_g(t)}(t)\big]. \tag{102}$$

The middle term is nonpositive because $\widehat{a}_g(t)$ maximizes $\widehat{u}_{ga}(t)$. The first and third terms are each bounded above by $\alpha_q \varepsilon$, which proves Eq. (98). Summing over T requests yields the cumulative statement. □

Proposition C.5 is the technical connection between the first appendix and the routing story. The structural analysis shows why an FWP-style proxy can enlarge the decision-relevant quality gap for fragile requests. The LP appendix then shows that if that proxy is calibrated within error ε, the induced routing policy loses at most $2\alpha_q \varepsilon$ utility per decision in the linearized workload-conditioned objective.

This appendix establishes the two claims needed in the main text. First, the reduced-form surrogate is a constructive first-order approximation of quantized MoE quality loss. Second, the LP-derived routing score can be derived as a regularized primal-dual step rather than as a disconnected scoring rule. Empirically, the signed score with backlog-based capacity prices drives the live router of Section 4.4, which serves mixed quality classes at $1.06\times$ to $1.20\times$ the throughput of the only static deployment that meets every class budget.

D Detailed Evaluation Protocol

Table 2: Per-component summary of evaluation metrics on Qwen3-30B-A3B and DeepSeek-V2-Lite. Bold marks the best per architecture for each metric. Multi-prefill routing rows aggregate over the 32-cell evaluation matrix at latency-only forward-pass cost.

Component	Metric	Qwen3-30B-A3B	DeepSeek-V2-Lite
Configuration prior (§4.1; 65 held-out configs)	Pearson	0.506	**0.807**
	Spearman	0.323	**0.807**
	Pairwise rank acc.	0.596	**0.822**
Request-level FWP (§4.2; top-20% recall)	same-instance $\Delta \log \mathrm{FWP}_{\mathrm{mean}}$	**0.97**	**0.98**
	activation-frequency baseline	0.12	0.27
Multi-prefill routing (§4.5; latency-only)	Best mean utility / 32 cells	**85.85**	**87.01**
	Best SLO compliance	**0.636**	**0.707**
	Δ utility vs. load-balanced	+2.95	+2.00

Stress-study metric. For request t routed to instance a,

$$U_t(a) = \alpha_s L_{g_t} - \alpha_q \widehat{\Delta Q}_t(a) - \beta\, \mathrm{Latency}_t(a) - \lambda_{\mathrm{SLO}}\, \mathbf{1}\left\{\widehat{\Delta Q}_t(a) > \tau_{g_t}\right\}, \tag{103}$$

and policies are compared by mean utility (higher is better) and SLO compliance.

Table 3: Cost-graded FWP-aware routing aggregated over the 32-cell evaluation matrix. Utility follows Eq. (103) (higher is better); bold marks the best per architecture for each metric.

	Qwen3-30B-A3B		DeepSeek-V2-Lite	
Policy	Util.	SLO comp.	Util.	SLO comp.
Latency-only mode cost				
3-prefill queue-aware	84.62	**0.636**	**87.01**	**0.707**
2-prefill (best, W_2-W_3)	**85.85**	0.621	86.30	0.653
1-prefill calibrated	83.81	0.567	84.33	0.621
Load-balanced	82.90	0.569	85.01	0.580
Per-step quality-optimal comparator	76.62	0.515	74.31	0.477
Shared-pool-capacity mode cost				
2-prefill (best, W_2-W_3)	29.33	0.116	69.51	0.348
Load-balanced	82.90	0.569	85.01	0.580

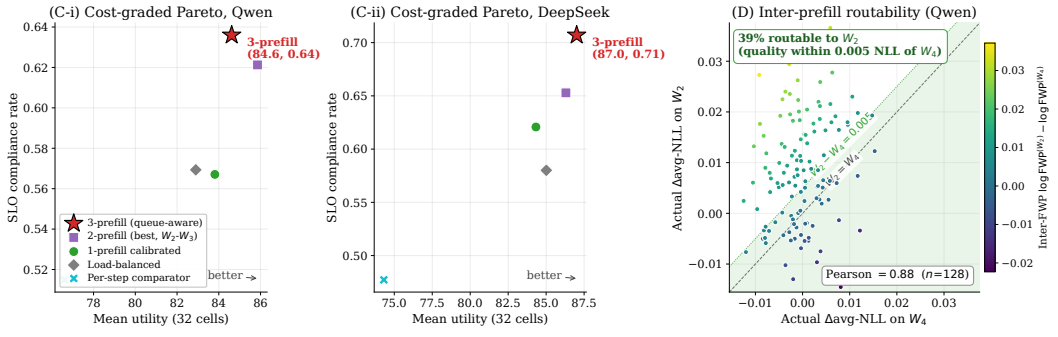


Figure 3: (C) Utility versus SLO compliance under latency-only prefill cost on Qwen and DeepSeek. (D) Qwen inter-FWP routability ($n = 128$): with one prefill on each of two instances, the inter-FWP delta finds that about 39% of requests are within 0.005 NLL of the highest-quality instance and can be served on the cheapest one. Both panels use the sampled-expert protocol (Section 4 Setup).

Model and data. The primary benchmark uses Qwen3-30B-A3B; DeepSeek-V2-Lite supplies the cross-architecture stress test. Expert fragility is measured on 128 experts sampled from four

representative MoE layers $\{6, 18, 30, 42\}$ via stratified activation-frequency and router-norm sampling. Each expert undergoes real HQQ leave-one-expert-out perturbations at $\{8, 4, 3, 2\}$-bit on 128 Wikitext-2 validation sequences (1024 tokens), yielding 512 perturbation runs. The LOEO data is used exclusively offline: it characterizes the fragility distribution, calibrates the free coefficients $\widehat{c}_e^{\text{free}}$, and supports the controlled cheap-predictor regression. The deployed proxy-aware policy reads measured request-level FWP signals at decision time, not the offline coefficients.

Configuration transfer. We evaluate 65 held-out quantization configurations over the sampled experts, comparing two monotone decay families (2^{-b} and 2^{-2b}) and two coefficient families (free and structured), keeping nonsampled experts at BF16. Figure 4 shows the resulting predicted-versus-actual scatter for the held-out 65-config task.

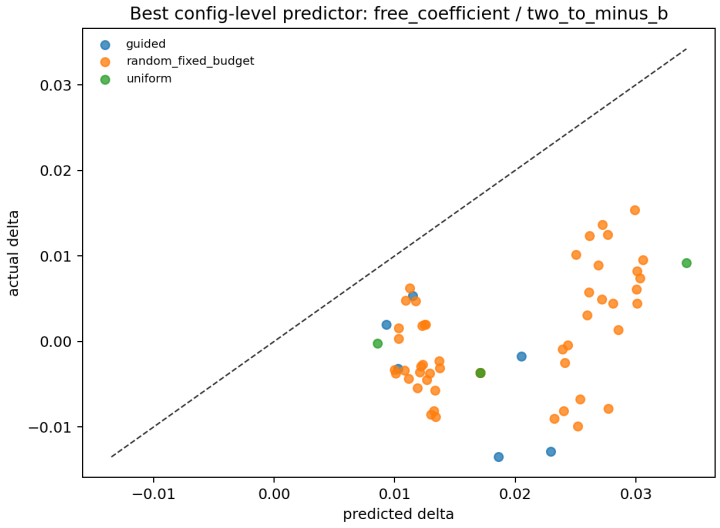


Figure 4: Configuration-level calibration scatter for the held-out 65-config transfer task on Qwen. The free-coefficient surrogate preserves weak ranking signal (Spearman 0.323), but predicted scale is poorly calibrated against measured quality loss, yielding negative held-out R^2. The deployable signal in §4.2 is the FWP family, not this surrogate.

Request-level evaluation. The serving pool is three $\{2, 3, 4\}$-bit instances under two source modes: a default Wikitext-2 mode (128 requests) and an extended real-prompt mode covering short QA (OpenBookQA, 32), code (MBPP, 32), and long-context (CNN/DailyMail, 24) buckets, totaling 648 request-instance rows with real service profiles.

Table 4: Complete-instance Qwen3-30B-A3B routing at a fixed quality budget (τ = W_4 mean loss $+0.10 = 0.1513$ nats/token; 88 extended requests; loss is mean ΔNLL from BF16). Multipliers follow the memory-bound $4/b$ decode model; parenthesized multipliers violate the budget (static W_2: ΔNLL 0.944, infeasible). Held-out: mean $\pm$ s.d. over 20 calibration/test splits.

Policy	$W_2/W_3/W_4$ (%)	Mean ΔNLL	Budget met	Multiplier	Held-out
Static W_4	0/0/100	0.051	Yes	1.000×	1.000×
Static W_3	0/100/0	0.183	No	(1.333×)	n/a
Prompt length	0/56/44	0.147	Yes	1.186×	1.183 ± 0.045
Activation frequency	0/59/41	0.150	Yes	1.197×	1.195 ± 0.038
FWP	0/85/15	0.151	Yes	**1.284×**	$\mathbf{1.282 \pm 0.045}$
True-loss ordering (reference)	0/91/9	0.149	Yes	1.303×	1.302 ± 0.023

Complete-instance fixed-pool comparison. For Table 4, all 6,144 expert blocks of Qwen3-30B-A3B are quantized to form the $W_2/W_3/W_4$ pool. The evaluation uses the 88 extended requests and defines quality loss as mean request-level Δavg-NLL from BF16. The population budget is $\tau = \overline{\Delta\text{NLL}}_{W4} + 0.10 = 0.1513$ nats/token. Requests are ordered by reference-instance FWP;

cutoffs over that population maximize $\mathcal{M} = \sum_{b\in\{2,3,4\}} p_b(4/b)$ subject to the mean quality budget, where p_b is the fraction assigned to the b-bit instance. The selected allocation is $p_2 = 0$, $p_3 = 0.8523$, and $p_4 = 0.1477$, giving $\mathcal{M} = 1.2841$ before rounding.

Proxy family. We compare ΔPPL, Δ average NLL, Δ rare-token-weighted NLL, $\Delta \log \mathrm{FWP}_{\mathrm{mean}}$, and $\Delta \log \mathrm{FWP}_{\mathrm{top1}}$. The audit target is request-level Δ average NLL; a request is labeled high-risk if its measured Δ avg-NLL falls in the top 20% of the held-out slice. Proxy criteria: normalized dynamic range, Spearman correlation, AUROC for high-risk detection, top-k recall, and held-out calibration error (with bootstrap confidence intervals).

Routing evaluation. Four calibrated load scenarios (burst_120, load_40, load_70, load_90) are crossed with utility-weight sweeps (q100/q250/q500 × l0.1/l0.5) under split latency/quality/combined SLO diagnostics, yielding the 32-cell evaluation matrix referenced in §4.5. Policies include fair all-pool baselines (measured fastest, static best quality, round robin, least loaded, latency only), activation-frequency baselines, coefficient-only routing, threshold policies, window-planner ablations, FWP-aware policies, and a per-step quality-optimal comparator. The deployment stress phase calibrates proxy scores to Δ avg-NLL with deterministic 60/20/20 train/validation/test splits and scores a cost-graded prefill family: 1-prefill calibrated (single prefill pass on the reference instance with cross-instance affine transfer), 2-prefill inter-FWP (one prefill on each of two instances with the inter-instance log-FWP delta as the proxy), and 3-prefill queue-aware (one prefill on each of the three pool instances with queue-aware dispatch). In the stress study each policy scores instances by the utility of Eq. (103) with fixed weights, that is, Eq. (12) with fixed prices, the predicted loss floored at zero, and a step SLO penalty, and the single-prefill policy reads FWP from the W_4 instance. Forward-pass acquisition is charged under two cost models: latency-only (the prefill pass extends request latency only) and shared-pool-capacity (the prefill pass competes with regular serving for the same GPU compute pool). Acceptance requires quality noninferiority, positive utility over latency-only, and positive throughput over static best quality.

DeepSeek-V2-Lite. The architecture differs in sparse-layer layout and routed-expert count. The diagnostic stack is replayed using 1024 LOEO perturbations, the 65-config ranking task, request-level FWP analysis, and the full cost-aware policy matrix to test which components generalize.

Routing pseudocode. The routing policy of Section 3.3 is summarized in Figure 5 for an arbitrary pool of instances; it places no restriction on their quantization configurations. Offline, per-instance loss predictors are fit on a labeled calibration sample and the window LP supplies the initial dual prices (lines 1 and 2). Online, one prefill pass on the reference instance measures FWP, the predictors transfer it to every instance, and the request goes to the argmax of Eq. (12) (lines 3 to 6). Dual prices update once per control window (line 7). The live pool of Section 4.4 (W_4, $\mathrm{W}_{4/8}$, W_8) and the offline studies (W_2, W_3, W_4) are both instances of this procedure.

Routing procedure **Calibrated single-pass routing**

Input: pool $\mathcal{A}$ of instances with arbitrary quantization configurations $\mathbf{b}^{(a)}$; reference instance a_{ref} (the cheapest instance, used for the FWP prefill); profiled service costs s_{ga}; per-class budgets τ_g; a labeled calibration sample; control window W

Offline (once per control window):

1. **Calibrate:** on the calibration sample, fit a loss predictor $\widehat{\ell}_a(x)$ for every $a \in \mathcal{A}$ from the prefill observables of a_{ref} (FWP and prompt NLL).
2. **Plan:** solve the window LP in Eq. (10) with $\widehat{\ell}$ to obtain dual prices (π_a, ν_g).

Online (per request x of class g):

3. **Prefill:** run x on $a_{\text{ref}} \to$ per-token NLL and routed experts $\to \mathrm{FWP}(x)$ (Eq. (7)).
4. **Transfer:** predict $\widehat{\ell}_a(x)$ for every $a \in \mathcal{A}$.
5. **Score:** $S_{ga} \leftarrow L_g - \pi_a s_{ga} - \nu_g\big(\widehat{\ell}_a(x) - \tau_g\big)$.
6. **Route:** serve x on $a^* \leftarrow \arg\max_{a \in \mathcal{A}} S_{ga}$.

Per control window:

7. Update dual prices: $\pi_a \leftarrow [\pi_a + \eta_\pi(\text{util}_a - C_a)]_+, \quad \nu_g \leftarrow [\nu_g + \eta_\nu(\bar{\widehat{\ell}}_g - \tau_g)]_+$.

Output: a routing decision a^* for every request.

Figure 5: Calibrated single-pass routing for an arbitrary pool of quantized instances, consolidating the policy of Section 3.3.

E Live Serving Protocol

Stack and instances. Qwen3-30B-A3B is served with vLLM 0.26 and online round-to-nearest group-wise quantization (group size 128), one instance per A100-40GB GPU, with maximum context 2,048 tokens and prefix caching disabled. W_4 and W_8 quantize every linear layer at 4 and 8 bits. $\mathrm{W}_{4/8}$ quantizes attention and router weights at 8 bits in every layer, expert weights at 8 bits in the 24 layers with the largest mean inverse router norm (layers 18 and 25 to 47), and expert weights at 4 bits elsewhere. FWP is computed from a prefill pass on a reference instance; static deployments use the same three serving GPUs as the pool.

Prompts and labels. We draw 150 prompts from each of UltraChat-200k, GSM8K, MBPP, and CNN/DailyMail ($\leq$ 1,400 prompt tokens), split them within each source into disjoint calibration and test halves of 300 prompts, and verify that the halves share no prompt. The reference text for each prompt is BF16's greedy response (up to 500 tokens). The label $\Delta\mathrm{NLL}_a(x)$ is the mean NLL of instance a on that response minus BF16's, measured through each serving endpoint and averaged over five repeats with shuffled co-batching; on the reported class means the standard error is at most 0.0002 nats/token. Mean test-half labels are 0.095 (W_4), 0.012 ($\mathrm{W}_{4/8}$), and 0.001 (W_8).

Router. Each run forms one control window: the window LP is solved once on the calibration half before the run, and the router takes the per-request argmax of Eq. (12) without re-solving it. Within the window, the capacity term follows the instance backlog and the quality price receives virtual-queue corrections around the LP dual, as follows. Per-instance loss predictors $\widehat{\ell}_a(x)$ are linear in FWP and the prompt NLL from the same prefill pass, selected by five-fold cross-validation on the calibration half. For each request of class g the router computes $S_{ga} = -(Q_a + 1)/c_a - \nu_g\,\widehat{\ell}_a(x)$, where Q_a is the number of in-flight requests on instance a and c_a its measured capacity, so that $(Q_a + 1)/c_a$ is the backlog form of the capacity price. The quality price follows the virtual queue $\nu_g \leftarrow [\nu_g + \eta(\widehat{\ell}_{a^*}(x) - \tau_g)]_+$, warm-started from the dual of the calibration-half LP. The LP's per-class quality constraint is enforced at dispatch: instance a is admissible for a class-g request only if the class's running mean predicted loss stays within τ_g (W_8 is always admissible). The step size $\eta = 100$ was selected by replaying the router on the calibration half. A closed-loop client keeps 128 concurrent requests in flight across the pool, replacing each completed request immediately; every request decodes exactly 500 tokens.

Latency. Including the scoring pass, median time to first token is 0.26 to 0.33 s for the FWP-routed pool on the four mixed-class workloads and 0.30 to 0.33 s for the static deployments.

Table 5: Single-class budget sweep on the live pool (test half, same load as Table 1). Throughput rises monotonically as the budget loosens, and every target is met.

Budget τ	0.002	0.005	0.010	0.020	0.035	0.050	0.075
Throughput (tok/s)	2,325	3,340	4,201	4,495	4,939	5,040	5,087
Realized ΔNLL	0.0018	0.0047	0.0098	0.0194	0.0327	0.0449	0.0480
Share $W_4/W_{4/8}/W_8$ (%)	0/10/90	0/40/60	4/59/37	18/50/33	37/38/25	45/35/20	46/34/20