CROSS-ENTROPY GUIDED ROUTING IN MIXTURE-OF-EXPERTS LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse mixture-of-experts (MoE) large language models scale model capacity by routing each token to a small subset of experts. Their routers are regularized with load balancing terms and learn affinity scores through the language-model objective. However, these objectives do not provide direct alignment between routing affinities and token-level error. We introduce token-error supervision for sparse routing in two forms. The first form predicts an error score per expert. The affinity-weighted aggregate of these scores is aligned to the next-token cross-entropy loss, while the individual scores attenuate affinity before top-K selection. The second directly aligns the router's affinities to the model's objective without requiring an additional head or inference-time modification. Both formulations use the Itakura–Saito divergence or an exponential negative log-likelihood for aligning affinities and token errors. Across two sparse MoE backbones and four multiple-choice question-answering benchmarks, we evaluate both supervision mechanisms. On Granite, our method improves accuracy by approximately 2.3 percentage points on average over a parameter-matched routing baseline. With stronger supervision, the gain on ARC-Challenge reaches 2.94 points. Both mechanisms preserve the native sparse execution budget and aggregation policy. Our code is available in the supplementary materials.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) models expand model capacity without proportionally increasing computation by routing each token to only K of N available experts (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). This conditional computation has become central to scaling modern language models (Jiang et al., 2024; Dai et al., 2024). Its effectiveness, however, depends on routing: for every token, the router determines which experts are executed and how their outputs contribute to the token representation. Standard sparse routers make this decision through affinity scores learned indirectly from the language-model objective and routing regularizers such as load balancing (Fedus et al., 2022). These scores provide relative preferences among experts, but they are not explicitly supervised against the model's token-level loss. Native MoE routing therefore lacks an explicit signal indicating whether the selected computation is likely to produce a high- or low-error prediction.

The realized next-token cross-entropy provides a direct token-level measure of prediction error for the routed computation. We study two ways of using this signal. The first, token-error supervision (TES), trains an error prediction head and aggregates the predicted errors based on affinity scores. Figure 1 compares standard sparse MoE routing with the two supervision mechanisms. The second, affinity-concentration supervision (ACS), uses the realized token loss to supervise the squared probabilities of the normalized selected-expert affinities, allowing training to sharpen or flatten the native routing distribution without an additional head or inference-time transformation. We supervise TES and ACS with the Itakura–Saito (IS) divergence (Itakura & Saito, 1968; Févotte et al., 2009). We further explore the exponential negative log-likelihood (ENLL) (Casella & Berger, 2002) as a simplified supervision scheme.

Our theoretical analysis shows that, for TES, both objectives train the aggregate prediction toward the same expected token-error target, but affect language-model optimization differently. When predicted and observed token error match, IS leaves the original cross-entropy (CE) update unchanged,

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

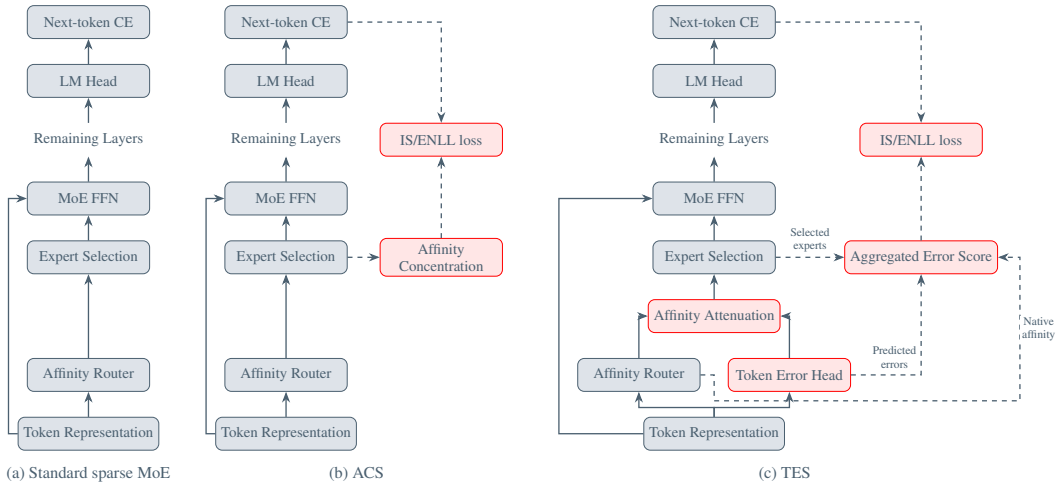


Figure 1: Overview of sparse MoE routing and the proposed supervision mechanisms: (a) the standard language-model pipeline; (b) affinity-concentration supervision (ACS); and (c) token-error supervision (TES) with affinity attenuation. MoE FFN denotes the routed expert computation and aggregation; routing supplies expert selection and aggregation weights. ACS computes concentration from normalized selected affinities. TES aggregates predicted errors using native affinities normalized over the executed experts. Dashed arrows denote auxiliary readout inputs, not stopped gradients.

whereas ENLL gives relatively greater gradient weight to lower-loss tokens. In TES, we subtract a scaled log-error score from native affinity before top-K selection, reducing the relative preference of experts with larger predicted error. In ACS, the same objectives instead sharpen or flatten native affinity according to the token loss. Section 3 develops the two mechanisms, while Appendix A analyzes their optima and gradients.

We evaluate TES and ACS on Granite 3.1 and OLMoE-1B across four multiple-choice question-answering (MCQA) benchmarks. On Granite, ACS-IS improves average accuracy by 2.1 percentage points over CE fine-tuning with frozen native affinity. TES-IS improves over a parameter-matched Dual Affinity baseline by 2.3 points on Granite and 0.5 points on OLMoE, averaged across all four benchmarks. With stronger supervision, its improvement on Granite ARC-Challenge reaches 2.94 points. These results show that directly supervising predicted token error or affinity concentration against the realized token loss can improve sparse routing, with the preferred mechanism depending on the backbone and task. Both methods preserve the native top-K execution budget and architecture-specific aggregation policy.

In summary, our contributions are as follows:

- We introduce two complementary ways to supervise sparse routing with realized next-token cross-entropy. A lightweight expert-indexed error head predicts token error through its native-affinity-weighted aggregate, while a headless alternative directly aligns native router outputs. We show that direct supervision enables token-error prediction and aligns affinity outputs with token error, without changing the sparse expert-execution budget.
- We further propose a method for integrating expert-indexed error scores into routing by subtracting a scaled log-error term from native affinity before top-K selection. The resulting route uses token-error information to rank experts while preserving the native execution budget and architecture-specific aggregation policy.
- We evaluate both supervision mechanisms against standard CE fine-tuning and a parameter-matched routing-head control across two sparse MoE backbones and four MCQA benchmarks, demonstrating accuracy improvements over the corresponding baselines.

2 Related Work

Sparse MoE routing and explicit supervision. Sparse MoE models use learned affinity to activate only the top-K experts, increasing capacity without proportional per-token computation (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). Later work changes how this sparse allocation is formed: Expert Choice lets experts select tokens (Zhou et al., 2022), while ReMoE replaces discontinuous top-K selection with differentiable ReLU routing (Wang et al., 2025).

A closer line of work explicitly guides router behavior. Expert-router coupling (ERC) binds router embeddings with expert capabilities through expert-specific proxy tokens (Lv et al., 2026), and Expert Divergence uses domain labels to encourage functional specialization (Li et al., 2026). Counterfactual analysis further shows that native routing can miss equal-compute alternatives with lower next-token loss (Yoon et al., 2026). Our method directly targets this gap through two complementary mechanisms. TES supervises an affinity-weighted prediction of routed token loss and uses the resulting expert-indexed error scores to refine selection without changing the top-K execution budget. ACS uses the same token-loss objectives to shape affinity without adding a prediction head.

Token difficulty and loss prediction. Learned loss prediction provides a general mechanism for estimating which inputs a model is likely to find difficult (Yoo & Kweon, 2019). At the query level, Hybrid LLM uses predicted difficulty to route requests between models of different capacities (Ding et al., 2024). Within MoE models, Huang et al. (2024) infer difficulty from router confidence and activate more experts for difficult inputs; DynaMoE derives token-difficulty labels from agreement between nested experts and the full-width MLP (Nishu et al., 2025); and Ada-K learns a token-dependent expert budget through reinforcement learning (Zhao et al., 2025). These methods use difficulty to allocate computation, typically by changing the capacity or number of experts assigned to each token. Our method instead uses realized next-token loss to supervise sparse language-model routing.

Uncertainty-aware routing. In sparse language models, recent work represents routing uncertainty through different probabilistic signals. Variational Mixture-of-Experts Routing (VMoER) (Li & Wicker, 2026) represents routing logits using input-dependent probability distributions and performs variational inference over the resulting routing decisions. Uncertainty-Aware Routing (UAR) (Chen et al., 2026) uses router entropy to adapt both expert capacity and routing regularization. Related probabilistic routers include Grassmannian MoE (Shihab et al., 2026), which controls routing through Bingham subspace geometry, while VI-MoLE (Saliencro et al., 2026) predicts counterfactual residual risk to allocate a variable budget among LoRA experts. Our method uses deterministic, fixed-budget top-K routing without sampling router distributions and introduces direct supervision from the routed model's next-token loss. Fixed-budget execution is also retained by VMoER's logit-space inference; our distinction is the directly supervised error signal.

In dense MoE models for time-series regression, MoGU (Aviv et al., 2025) provides the closest conceptual precedent for using an expert-specific predictive-error signal to control mixture weights. It models each regression expert as a Gaussian predictor, trains its variance through Gaussian negative log-likelihood, and aggregates expert predictions using normalized inverse variance. Our formulation differs in both target and routing semantics: the error head does not define a Gaussian expert likelihood or estimate calibrated predictive variance. Instead, native affinity aggregates the executed experts' positive scores into a token-error prediction supervised against next-token cross-entropy; the scaled log-error values subsequently reduce the relative affinity of experts with larger predicted error before selection.

3 Method

We introduce two mechanisms for aligning sparse MoE routing with next-token cross-entropy loss. We first review sparse MoE routing in Section 3.1. Next, Sections 3.2–3.3 present token-error supervision (TES), and Section 3.4 introduces affinity-concentration supervision (ACS). Finally, Section 3.5 details the overall training objective and its computational cost.

3.1 SPARSE MOE ROUTING

Sparse mixture-of-experts models increase parameter capacity while limiting per-token computation by evaluating only K of N routed experts in each MoE layer (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). A learned router assigns each token to its active experts and determines their contributions to the layer output. Consider a token representation $\mathbf{h}_t \in \mathbb{R}^d$. Omitting the layer index for clarity, the affinity router computes

$$\mathbf{a}_t = \mathbf{W}_{\text{aff}}\mathbf{h}_t + \mathbf{b}_{\text{aff}}, \qquad \mathbf{p}_t = \text{softmax}(\mathbf{a}_t), \qquad \mathcal{S}_t = \text{TopK}(\mathbf{p}_t, K). \tag{1}$$

Here $\mathbf{W}_{\text{aff}} \in \mathbb{R}^{N\times d}$ and $\mathbf{b}_{\text{aff}} \in \mathbb{R}^N$ are the affinity-router parameters, while $\mathbf{a}_t, \mathbf{p}_t \in \mathbb{R}^N$ are the affinity logits and probabilities, respectively. The routed expert branch evaluates the experts in $\mathcal{S}_t$ and combines their outputs as

$$\mathbf{y}_t = \sum_{i\in\mathcal{S}_t} g_{t,i} E_i(\mathbf{h}_t), \tag{2}$$

where $g_{t,i}$ denotes the weight derived from $\mathbf{p}_t$. Depending on the model, these weights may be renormalized over $\mathcal{S}_t$ or may preserve the selected affinity mass.

The affinity router is learned through the language-model objective and auxiliary routing objectives, such as load balancing (Shazeer et al., 2017; Fedus et al., 2022). These objectives make affinity effective for selecting and combining experts, but do not explicitly supervise it against the model's token-level error. We therefore retain affinity as the expert-preference signal and complement it with a separately supervised token-error signal, introduced in the following subsection.

3.2 TOKEN-ERROR HEAD AND AFFINITY ATTENUATION

TES augments the native router with a lightweight token-error head. From the shared pre-expert token representation $\mathbf{h}_t$, the head predicts one positive error score for each expert before expert selection:

$$\widehat{\mathbf{e}}_t = \text{softplus}(\mathbf{W}_e\mathbf{h}_t + \mathbf{b}_e), \qquad \widehat{\mathbf{e}}_t \in \mathbb{R}^N_{>0}. \tag{3}$$

Here $\mathbf{W}_e \in \mathbb{R}^{N\times d}$ and $\mathbf{b}_e \in \mathbb{R}^N$ are the error-head parameters. The component $\widehat{e}_{t,i}$ is the predicted error score for expert i at token t; positivity ensures that the logarithms used below are well defined.

Let $\mathcal{L}_{\text{CE},t}$ denote the realized next-token cross-entropy, and let $\mathcal{S}_t^{\text{act}}$ denote the experts executed in the same forward pass. Thus, $\mathcal{S}_t^{\text{act}} = \mathcal{S}_t$ under native routing and $\mathcal{S}_t^{\text{act}} = \widetilde{\mathcal{S}}_t$ under the error-aware attenuation defined below. Because this loss evaluates the complete routed prediction rather than an individual expert, we form a single affinity-weighted prediction over the active experts:

$$\bar{p}_{t,i} = \frac{p_{t,i}}{\sum_{j\in\mathcal{S}_t^{\text{act}}} p_{t,j}}, \quad i \in \mathcal{S}_t^{\text{act}}, \qquad \widehat{e}_t = \sum_{i\in\mathcal{S}_t^{\text{act}}} \bar{p}_{t,i}\widehat{e}_{t,i}. \tag{4}$$

Equation 4 forms the token-error prediction supervised in Section 3.3; active experts with larger native affinity contribute more to the predicted error.

We use the expert-indexed error predictions to adjust the native affinity logits $a_{t,i}$, defined in Equation 1, before expert selection:

$$\widetilde{a}_{t,i} = a_{t,i} - \gamma\phi_\tau(\widehat{e}_{t,i}), \qquad \widetilde{\mathbf{p}}_t = \text{softmax}(\widetilde{\mathbf{a}}_t), \qquad \widetilde{\mathcal{S}}_t = \text{TopK}(\widetilde{\mathbf{p}}_t, K). \tag{5}$$

Here,

$$\phi_\tau(\widehat{e}_{t,i}) = \log\left(1 + \frac{\widehat{e}_{t,i}}{\tau}\right), \qquad \tau > 0, \tag{6}$$

is the started logarithm of the predicted error (Rocke & Durbin, 2003). The reference scale τ controls the transition between small and large predicted errors, while $\gamma \geq 0$ controls the strength of error attenuation relative to native affinity. We set $\gamma = 1$ and $\tau = 1$ by default. FFN aggregation uses $\widetilde{\mathbf{p}}_t$ under the architecture's native selected-weight normalization policy; the auxiliary error readout in Equation 4 instead uses native affinities.

Exponentiating the adjusted logits factorizes each softmax term into its native affinity and an inverse-error factor, giving

$$\widetilde{p}_{t,i} = \frac{p_{t,i}\,(1 + \widehat{e}_{t,i}/\tau)^{-\gamma}}{\sum_{j=1}^N p_{t,j}\,(1 + \widehat{e}_{t,j}/\tau)^{-\gamma}}. \tag{7}$$

Because ϕ_τ is increasing, experts with larger predicted error receive lower adjusted logits. Appendix A.4 provides the full derivation of Equation 7. Appendix A.5 compares the started-log transformation with the scale-invariant pure-log alternative.

3.3 TOKEN-ERROR SUPERVISION

We align $\widehat{e}_t$ against the realized token loss using the Itakura–Saito (IS) divergence, a scale-invariant measure of relative disagreement between positive quantities (Itakura & Saito, 1968; Févotte et al., 2009). Because IS requires a positive observation, we apply a small numerical floor:

$$\mathcal{L}^{+}_{\mathrm{CE},t} = \max(\mathcal{L}_{\mathrm{CE},t}, \varepsilon_{\mathrm{IS}}), \qquad \mathcal{L}_{\mathrm{IS},t} = \frac{\mathcal{L}^{+}_{\mathrm{CE},t}}{\widehat{e}_t} - \log\left(\frac{\mathcal{L}^{+}_{\mathrm{CE},t}}{\widehat{e}_t}\right) - 1. \tag{8}$$

Exponential negative log-likelihood (ENLL) provides an alternative supervision objective with the same prediction target but different language-model gradients. It treats the nonnegative token loss as an observation from an exponential distribution with conditional mean $\widehat{e}_t$:

$$\mathcal{L}_{\mathrm{ENLL},t} = \frac{\mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t} + \log \widehat{e}_t. \tag{9}$$

Through the aggregate error prediction in Equation 4, the individual expert scores receive direct gradients in proportion to their normalized affinities, but are not identified as counterfactual expert losses. We therefore interpret them as learned error-aware routing signals. In both TES and ACS, the observed token CE remains attached to the computation graph, allowing gradients through both the prediction or routing statistic and the observed loss. Appendices A.2 and A.3 analyze their gradient allocation and the resulting language-model updates.

Both objectives align the aggregate prediction with its supervised token loss. Across the contextual occurrences c of a fixed target token identity t', the supervision target is the expected realized token loss. The corresponding context-averaged token-identity optima are

$$\widehat{e}^{\star}_{\mathrm{IS},t'} = \mathbb{E}_{c|t=t'}[\mathcal{L}^{+}_{\mathrm{CE},t,c}], \qquad \widehat{e}^{\star}_{\mathrm{ENLL},t'} = \mathbb{E}_{c|t=t'}[\mathcal{L}_{\mathrm{CE},t,c}]. \tag{10}$$

Here, $\widehat{e}^{\star}_{\mathrm{IS},t'}$ and $\widehat{e}^{\star}_{\mathrm{ENLL},t'}$ are scalar reference optima obtained by minimizing the context-averaged supervision loss with respect to a single prediction for target token identity t'. The implemented head remains context dependent; these reference values do not require identical predictions across occurrences. Appendix A.1 gives the complete derivation of Equation 10.

3.4 AFFINITY-CONCENTRATION SUPERVISION

Affinity-concentration supervision (ACS) provides an alternative to the learned token-error head. We directly align the probabilities by replacing the error score $\widehat{e}_{t,i}$ in Equation 4, with the normalized native affinity $\bar{p}_{t,i}$:

$$C_t = \sum_{i \in \mathcal{S}_t} \bar{p}_{t,i}\bar{p}_{t,i} = \sum_{i \in \mathcal{S}_t} \bar{p}^2_{t,i} \qquad \frac{1}{K} \le C_t \le 1. \tag{11}$$

where C_t denotes the affinity concentration. Its inverse, $1/C_t$, is the Hill effective number of selected experts, while $-\log C_t$ is the corresponding order-two Rényi entropy (Rényi, 1961; Hill, 1973). This gives C_t a direct routing interpretation: larger C_t corresponds to a smaller effective number of experts carrying the selected affinity mass, whereas smaller C_t corresponds to a more evenly distributed route. In particular, $C_t = 1/K$ under uniform selected affinity and approaches one when affinity concentrates on a single expert.

ACS applies the token-error supervision objectives introduced in Section 3.3, replacing the aggregate prediction $\widehat{e}_t$ in Equations 8 and 9 with C_t. Here, C_t is a bounded routing statistic, not an unrestricted token-loss prediction. Their direct gradients with respect to concentration are

$$\frac{\partial \mathcal{L}_{\text{ACS-IS},t}}{\partial C_t} = \frac{C_t - \mathcal{L}^{+}_{\mathrm{CE},t}}{C_t^2}, \tag{12}$$

$$\frac{\partial \mathcal{L}_{\text{ACS-ENLL},t}}{\partial C_t} = \frac{C_t - \mathcal{L}_{\mathrm{CE},t}}{C_t^2}. \tag{13}$$

When the IS floor is inactive, the two objectives produce the same direct concentration gradient. This gradient favors sharper selected affinity when token loss exceeds concentration and flatter affinity otherwise; the complete parameter update also depends on other gradient paths. ACS therefore uses token error to supervise routing affinities, without adding an error head or applying an error-aware routing transformation at inference. Appendix A.9 gives its bounded optimum, router-logit gradients, and additional analysis.

3.5 Overall Training Objective and Computational Scope

Restoring the layer indices omitted above, let $\mathcal{B}_{\text{tok}}$ denote the supervised token positions in a mini-batch, excluding padding and ignored targets, and let $\mathcal{M}$ denote the MoE layers receiving supervision. Let $m \in \{\text{TES-IS}, \text{TES-ENLL}, \text{ACS-IS}, \text{ACS-ENLL}\}$ denote the active supervision objective, and let $\mathcal{L}_{\text{CE}}$ denote the cross-entropy averaged over $\mathcal{B}_{\text{tok}}$. We average the active objective over supervised tokens and layers as

$$\mathcal{L}_m = \frac{1}{|\mathcal{M}||\mathcal{B}_{\text{tok}}|} \sum_{l \in \mathcal{M}} \sum_{t \in \mathcal{B}_{\text{tok}}} \mathcal{L}_{m,t}^{(l)}. \tag{14}$$

Here $\mathcal{L}_{m,t}^{(l)}$ is evaluated using the learned aggregate prediction $\widehat{e}_t^{(l)}$ when m is a TES objective, or the affinity concentration $C_t^{(l)}$ when m is an ACS objective. The complete training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_m \mathcal{L}_m + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}. \tag{15}$$

Here, $\mathcal{L}_{\text{aux}}$ is the architecture-native router regularizer, typically encouraging balanced expert usage (Shazeer et al., 2017; Fedus et al., 2022); $\lambda_m, \lambda_{\text{aux}} \geq 0$ are regularization weights of $\mathcal{L}_m$ and $\mathcal{L}_{\text{aux}}$, respectively. Each MoE layer equipped with TES includes an additional linear projection with $N(d+1)$ parameters and $\mathcal{O}(Nd)$ operations per token, while preserving the native top-K execution budget. ACS adds no parameters and retains native routing at inference. For MCQA adaptation, the primary task term is answer-choice CE, while the auxiliary target remains next-token CE (Appendix A.7).

4 Experiments and Results

We evaluate whether aligning MoE routing with token loss improves downstream accuracy relative to parameter-matched controls under a common supervision setting (Section 4.2). We then analyze whether predicted error tracks observed cross-entropy loss (Section 4.3). Finally, we ablate the number of supervised MoE layers (Section 4.4).

4.1 Experimental Setting

We evaluate Granite 3.1 3B-A800M and OLMoE-1B-7B-SFT (IBM Granite Team, 2024; Muennighoff et al., 2024) on ARC-Challenge, OpenBookQA, SciQ, and MedMCQA (Clark et al., 2018; Mihaylov et al., 2018; Welbl et al., 2017; Pal et al., 2022). Within each backbone and dataset, all configurations use the same registered train, validation, and test splits and the same answer-text multiple-choice protocol without in-context examples. Candidate answers are ranked by their mean conditional token log-probability. We report accuracy on a fixed 500-example test set; split construction and prompting are documented in Appendix A.7.

Where applicable, we follow VMoER's Stage-1 MAP adaptation setting (Li & Wicker, 2026): the same dataset suite and target split sizes, three-epoch LoRA adaptation of attention Q/K/V and routed experts, AdamW, and the Granite learning rate, schedule, warmup, and effective batch size. Because VMoER does not release split identities, we construct deterministic partitions targeting the reported sizes and filter invalid examples. Our answer-text candidate scoring differs from VMoER's generated-letter protocol, and the OLMoE recipe is adapted to that backbone.

Both models are adapted for three epochs using rank-8 LoRA with $\alpha = 8$ and dropout 0.05. Granite uses a learning rate of 10^{-4} and effective batch size 16, whereas OLMoE uses 2×10^{-5} and batch size 8. We evaluate training and initialization seeds 42, 43, and 45 for the primary accuracy comparison. Method-specific supervision is applied at the final sparse MoE layer in the primary

Table 1: Test answer-choice accuracy (%) for Granite 3.1 3B-A800M and OLMoE-1B-7B-SFT across four MCQA benchmarks. Entries are mean ± sample standard deviation over seeds 42, 43, and 45. Within each backbone, rows form three parameter-matched comparison groups: CE versus ACS with frozen affinity, CE versus ACS with affinity tuning, and Dual Affinity versus TES. All entries use epoch-3 checkpoints except the OLMoE affinity-tuned group, which uses validation-NLL-selected checkpoints. Bold marks the highest mean within each backbone, benchmark, and comparison group. † SciQ is near saturation.

Model	Method	ARC-Challenge	OpenBookQA	SciQ†	MedMCQA
Granite 3.1	CE	65.33 ± 2.01	71.67 ± 1.01	88.40 ± 0.53	39.60 ± 9.57
	ACS-ENLL (Ours)	67.40 ± 0.00	72.60 ± 0.53	89.27 ± 0.81	42.80 ± 0.40
	ACS-IS (Ours)	$\mathbf{67.87 \pm 0.76}$	$\mathbf{72.73 \pm 0.95}$	$\mathbf{89.87 \pm 0.70}$	$\mathbf{42.87 \pm 0.76}$
	CE + affinity tuning	67.53 ± 1.14	$\mathbf{74.00 \pm 0.72}$	88.60 ± 0.40	41.73 ± 1.22
	ACS-ENLL + affinity tuning (Ours)	69.87 ± 1.33	73.53 ± 0.90	89.20 ± 0.92	$\mathbf{42.13 \pm 1.27}$
	ACS-IS + affinity tuning (Ours)	$\mathbf{70.60 \pm 1.80}$	73.93 ± 0.31	$\mathbf{89.53 \pm 0.42}$	40.67 ± 1.45
	Dual Affinity	65.33 ± 2.00	71.27 ± 1.86	88.27 ± 0.31	37.40 ± 7.79
	TES-ENLL (Ours)	65.53 ± 0.42	72.27 ± 0.31	$\mathbf{88.53 \pm 1.67}$	$\mathbf{43.67 \pm 1.81}$
	TES-IS (Ours)	$\mathbf{66.40 \pm 0.35}$	$\mathbf{73.00 \pm 0.53}$	88.33 ± 0.64	$\mathbf{43.67 \pm 1.10}$
OLMoE	CE	$\mathbf{63.27 \pm 0.31}$	$\mathbf{70.40 \pm 0.40}$	88.80 ± 0.40	$\mathbf{44.93 \pm 0.70}$
	ACS-ENLL (Ours)	62.93 ± 0.64	70.20 ± 0.35	88.73 ± 0.23	44.53 ± 0.76
	ACS-IS (Ours)	62.73 ± 0.31	69.93 ± 0.46	$\mathbf{88.87 \pm 0.12}$	44.20 ± 0.53
	CE + affinity tuning	63.53 ± 0.58	$\mathbf{71.87 \pm 0.42}$	88.80 ± 0.20	44.47 ± 0.46
	ACS-ENLL + affinity tuning (Ours)	$\mathbf{63.80 \pm 0.53}$	71.00 ± 0.60	88.87 ± 0.42	44.27 ± 0.76
	ACS-IS + affinity tuning (Ours)	63.60 ± 0.35	69.33 ± 2.42	$\mathbf{89.07 \pm 0.31}$	$\mathbf{44.53 \pm 1.17}$
	Dual Affinity	61.27 ± 1.40	70.80 ± 1.06	$\mathbf{89.80 \pm 0.69}$	43.93 ± 0.76
	TES-ENLL (Ours)	62.93 ± 0.12	69.87 ± 0.61	88.80 ± 0.35	43.87 ± 0.81
	TES-IS (Ours)	$\mathbf{63.53 \pm 0.46}$	$\mathbf{70.93 \pm 0.23}$	88.80 ± 0.40	$\mathbf{44.53 \pm 0.12}$

comparison. Native affinity parameters remain frozen except in rows explicitly marked "+ affinity tuning," and every configuration preserves the native expert-execution budget.

For a fair evaluation, we compare ACS and TES with parameter-matched baselines under the IS and ENLL alignment schemes. ACS is compared with the CE fine-tuning baseline, while TES is compared with Dual Affinity, which adds an affinity head with the same parameter count and averages the two router outputs. To match the TES configuration, the original pretrained router is kept frozen and only the second affinity head is updated during fine-tuning. For each configuration, we report the mean and standard deviation across three seeds. Unless stated otherwise, both methods use a supervision coefficient of $\lambda_m = 10^{-3}$. Validation-NLL-selected checkpoints and complete training configurations are described in Appendix A.7. TES sensitivity to the error-head learning rate and attenuation scale is reported in Appendix A.6.

4.2 Downstream Accuracy

Table 1 compares TES and ACS across four datasets and two models. TES-IS improves ARC-Challenge accuracy over Dual Affinity by 1.07 percentage points on Granite and 2.26 points on OLMoE. It also improves OpenBookQA by 1.73 and 0.13 points, respectively, and OLMoE MedMCQA by 0.60 points. TES-ENLL improves ARC-Challenge on both backbones, although by smaller amounts. The objective ordering varies across tasks: on Granite SciQ, ENLL gives the larger TES gain of 0.26 points, whereas both TES objectives fall 1.00 point below Dual Affinity on OLMoE SciQ.

ACS produces its clearest gains on Granite. With native affinity frozen, ACS-IS improves over CE by 2.54 points on ARC-Challenge, 1.06 on OpenBookQA, and 1.47 on SciQ. Allowing affinity tuning changes the reference comparison: against affinity-tuned CE, ACS-IS improves ARC-Challenge by 3.07 points and SciQ by 0.93 points, while ACS-ENLL improves MedMCQA by 0.40 points. On OLMoE, ACS gains are smaller and task-dependent, and neither objective improves OpenBookQA in either affinity setting. These results distinguish the benefit of concentration supervision from that of simply unfreezing the router.

Table 2 evaluates stronger supervision by increasing λ_m to 10^{-2}, reporting both accuracy and changes from the corresponding result in Table 1. Stronger supervision improves accuracy on ARC-

Table 2: Effect of increasing the method-supervision coefficient to $\lambda_m = 10^{-2}$. Each cell reports epoch-3 test accuracy (%) followed in brackets by its change from the corresponding $\lambda_m = 10^{-3}$ method result, in percentage points. Bold indicates higher mean accuracy than the corresponding $\lambda_m = 10^{-3}$ configuration.

Model	Mechanism	Objective	ARC-C	OpenBookQA	SciQ	MedMCQA
Granite	TES, frozen	IS	**68.27** [+1.87]	72.47 [−0.53]	**89.20** [+0.87]	43.67 [+0.00]
		ENLL	**67.73** [+2.20]	**72.73** [+0.47]	**90.13** [+1.60]	43.40 [−0.27]
Granite	ACS, trainable	IS	**71.20** [+0.60]	**74.53** [+0.60]	88.80 [−0.73]	**41.07** [+0.40]
		ENLL	**70.47** [+0.60]	**75.20** [+1.67]	89.00 [−0.20]	41.47 [−0.66]
OLMoE	ACS, frozen	IS	**63.40** [+0.67]	69.80 [−0.13]	**89.67** [+0.80]	44.07 [−0.13]
		ENLL	**63.93** [+1.00]	69.20 [−1.00]	**90.07** [+1.34]	44.13 [−0.40]

Challenge. Granite TES-IS reaches 68.27%, an increase of 1.87 points over the corresponding 10^{-3} result and 2.94 points over Dual Affinity. Granite TES-ENLL reaches 90.13% on SciQ, improving by 1.60 points over its smaller-coefficient result and by 1.86 points over its control. Stronger supervision also benefits ACS. Granite affinity-tuned ACS-ENLL improves OpenBookQA by 1.67 points relative to its 10^{-3} setting.

These additional gains do not imply that the larger coefficient is preferable for every task. For example, Granite TES-IS loses 0.53 points on OpenBookQA, and both Granite affinity-tuned ACS objectives lose accuracy on SciQ. We therefore retain the common coefficient in Table 1 and report the complete paired changes in Table 2, rather than selecting the better coefficient separately for each test benchmark. Together, the comparisons show accuracy improvements under a shared setting and further gains from stronger supervision on several tasks.

4.3 Error-Prediction Analysis

Figure 2 examines whether the aggregate prediction $\widehat{e}_t$ correlates with observed next-token NLL across layer–token-identity groups, after averaging contextual occurrences within each group. Each group is defined by a routed layer ℓ and target token identity t'. Within each seed, we average predicted and observed errors over the contextual sequences whose next-token target is t'. We then average the group means across seeds and form ten equal-count bins by predicted error. This is the layer-specific empirical counterpart of the context averaging in Equation 10. In the figure, we standardize both quantities.

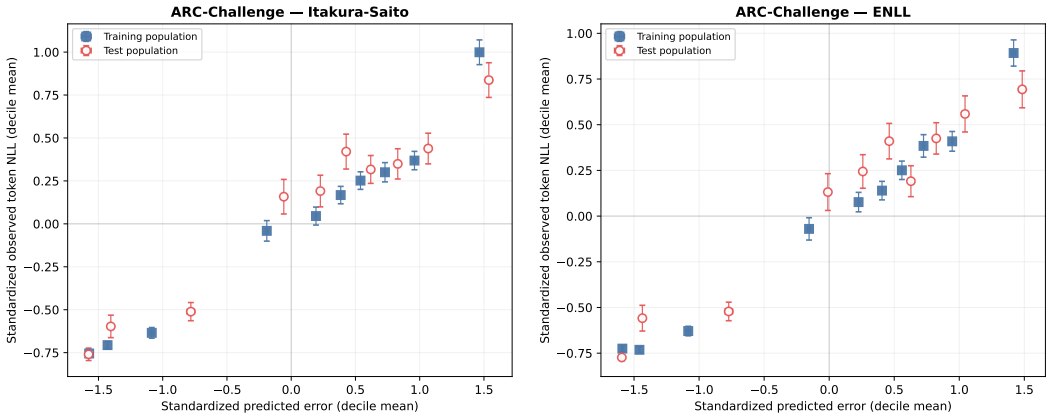
Both objectives capture token difficulty. Across test-set layer–token-identity groups, Pearson/Spearman correlations are $0.452/0.564$ for IS and $0.449/0.562$ for ENLL. Observed NLL increases across eight of the nine adjacent-decile transitions for both objectives. These standardized plots assess association and out-of-sample shift, rather than absolute NLL calibration. Appendix A.8 presents the corresponding OpenBookQA, SciQ, and MedMCQA analyses.

4.4 Ablation Studies

Table 3 compares the primary final-layer setting with all-layer ACS on OLMoE while keeping native affinity frozen and the supervision coefficient fixed. Extending ACS to all 16 MoE layers changes accuracy by at most 0.46 percentage points across objectives and datasets, with neither scope consistently performing better; final-layer supervision therefore remains the simpler competitive default. Additional ablations of the error-to-routing transformation, TES hyperparameters, supervision strength, affinity-router fine-tuning across layer scopes, and depth-dependent coefficient scaling are provided in Appendices A.5, A.6, A.10, A.11, and A.13.

5 Conclusion

Summary. We presented two complementary mechanisms for aligning sparse MoE routing with token-level loss. TES predicts expert-error scores that guide affinity attenuation, while ACS directly aligns native affinity concentration without an additional head. Across two MoE backbones and four MCQA benchmarks, both mechanisms yield accuracy gains over their corresponding controls while



(a) Itakura–Saito **(b) ENLL**

Figure 2: Aggregate token-error prediction on ARC-Challenge for Granite TES with started-log attenuation. The horizontal and vertical axes show standardized predicted error and observed token NLL, respectively; both use training-set moments. Points are equal-count decile means over layer–token-identity groups after averaging contextual occurrences within each group and then averaging group means over seeds 42–44. Whiskers show ± 1 standard error across groups. Blue squares and red circles denote the training and test populations, respectively.

Table 3: OLMoE epoch-3 test accuracy (%; mean $\pm$ sample standard deviation over seeds 42, 43, and 45) for final-layer and all-layer ACS with frozen native affinity and $\lambda_m = 10^{-3}$. Bold marks the higher mean within each objective and benchmark.

Objective	Supervised MoE layers	ARC-Challenge	OpenBookQA	SciQ	MedMCQA
ACS-ENLL	Final layer	62.93 ± 0.64	$\mathbf{70.20 \pm 0.35}$	88.73 ± 0.23	$\mathbf{44.53 \pm 0.76}$
ACS-ENLL	All 16 layers	$\mathbf{63.13 \pm 0.42}$	69.80 ± 0.40	$\mathbf{88.80 \pm 0.40}$	44.07 ± 0.81
ACS-IS	Final layer	62.73 ± 0.31	69.93 ± 0.46	88.87 ± 0.12	$\mathbf{44.20 \pm 0.53}$
ACS-IS	All 16 layers	$\mathbf{63.07 \pm 0.50}$	$\mathbf{70.13 \pm 0.12}$	$\mathbf{89.07 \pm 0.31}$	44.00 ± 0.35

preserving the native sparse execution budget and aggregation policy. Stronger supervision provides additional gains, with TES-IS reaching a 2.94-percentage-point improvement over Dual Affinity on Granite ARC-Challenge. These findings establish direct token-loss supervision as a useful complement to affinity-based routing and highlight supervision strength and depth as important design choices.

Limitations and Future Work. Our evaluation covers two sparse MoE backbones, four multiple-choice question-answering benchmarks, fixed data splits, and three training seeds. Larger-scale models and pretraining-scale optimization are outside the scope of this evaluation. Finally, although both methods preserve the native top-K expert budget, we do not measure end-to-end latency or memory overhead from the additional TES projection. Future work will extend our approach to pretraining paradigms and to additional backbones and datasets.

AI USE STATEMENT

AI tools were used to assist with text editing, manuscript organization, LaTeX preparation, and the identification of potential mistakes and inconsistencies. All suggestions, citations, and scientific claims were reviewed and verified by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

Gilad Aviv, Jacob Goldberger, and Yoli Shavit. MoGU: Mixture-of-gaussians with uncertainty-based gating for time series forecasting. *arXiv preprint arXiv:2510.07459*, 2025. URL https://arxiv.org/abs/2510.07459.

George Casella and Roger L. Berger. *Statistical Inference*. Duxbury, Belmont, CA, 2 edition, 2002. ISBN 9780534243128. URL https://ci.nii.ac.jp/ncid/BA5282379X.

Yilong Chen, Junyuan Shang, Yuchen Feng, Zhenyu Zhang, Naibin Gu, Ziqi Wang, Tingwen Liu, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. Uncertainty-aware routing for principled alignment with MoE dynamics. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 38865–38880, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1801. URL https://aclanthology.org/2026.acl-long.1801/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

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. *arXiv preprint arXiv:2401.06066*, 2024. URL https://arxiv.org/abs/2401.06066.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2404.14618.

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

Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the Itakura–Saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, Mar 2009. doi: 10.1162/neco.2008.04-08-771. URL https://doi.org/10.1162/neco.2008.04-08-771.

M. O. Hill. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2):427–432, Mar 1973. doi: 10.2307/1934352. URL https://doi.org/10.2307/1934352.

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

IBM Granite Team. Granite-3.1-3B-A800M-Base. Hugging Face model card, 2024. URL https://huggingface.co/ibm-granite/granite-3.1-3b-a800m-base.

Fumitada Itakura and Shuzo Saito. Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the 6th International Congress on Acoustics*, pp. C17–C20, 1968. URL https://cir.nii.ac.jp/crid/1570854175842518528.

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

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.16668.

Albus Yizhuo Li and Matthew Wicker. Variational routing: A scalable bayesian framework for calibrated mixture-of-experts transformers. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.09453.

Jiaang Li, Haibin Chen, Langming Liu, Yujin Yuan, Yadao Wang, Yizhen Zhang, Chengting Yu, Xin Tong, Weidong Zhang, Shilei Liu, Wenbo Su, and Bo Zheng. Expert divergence learning for MoE-based language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.00054.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. Coupling experts and routers in mixture-of-experts via an auxiliary loss. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 75251–75271, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/79ff18412c1a5816f071a796375abc3d-Abstract-Conference.html.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018. URL https://arxiv.org/abs/1809.02789.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024. URL https://arxiv.org/abs/2409.02060.

Kumari Nishu, Sachin Mehta, Samira Abnar, Mehrdad Farajtabar, Maxwell Horton, Mahyar Najibi, Moin Nabi, Minsik Cho, and Devang Naik. From dense to dynamic: Token-difficulty driven MoEfication of pre-trained LLMs. *arXiv preprint arXiv:2502.12325*, 2025. URL https://arxiv.org/abs/2502.12325.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. *arXiv preprint arXiv:2203.14371*, 2022. URL https://arxiv.org/abs/2203.14371.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 547–561. University of California Press, 1961. URL https://cir.nii.ac.jp/crid/1571417126122876416.

David M. Rocke and Blythe Durbin. Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, May 2003. doi: 10.1093/bioinformatics/btg107. URL https://doi.org/10.1093/bioinformatics/btg107.

Tom Saliencro, Rohan Desai, Priya Nair, Maya Lindqvist, and Daniel Whitmore. Uncertainty is not enough: Value-of-information routing for mixtures of LoRA experts. *arXiv preprint arXiv:2608.02528*, 2026. URL https://arxiv.org/abs/2608.02528.

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. *arXiv preprint arXiv:1701.06538*, 2017. URL https://arxiv.org/abs/1701.06538.

Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. Grassmannian mixture-of-experts: Concentration-controlled routing on subspace manifolds. *arXiv preprint arXiv:2602.17798*, 2026. URL https://arxiv.org/abs/2602.17798.

Ziteng Wang, Jun Zhu, and Jianfei Chen. ReMoE: Fully differentiable mixture-of-experts with ReLU routing. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.14711.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-4413. URL https://aclanthology.org/W17-4413/.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 93–102, June 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Yoo_Learning_Loss_for_Active_Learning_CVPR_2019_paper.html.

Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. When are experts misrouted? counterfactual routing analysis in mixture-of-experts language models. *arXiv preprint arXiv:2605.07260*, 2026. URL https://arxiv.org/abs/2605.07260.

Zijia Zhao, Longteng Guo, Jie Cheng, Xuange Gao, Hua Huang, and Jing Liu. Ada-K routing: Boosting the efficiency of MoE-based LLMs. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 89619–89635, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/df22a19686a558e74f038e6277a51f68-Abstract-Conference.html.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc V Le, and James Laudon. Mixture-of-experts with expert choice routing. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35*, volume 35, pp. 7103–7114. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-0515. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html.

A Appendix

A.1 Scalar Loss Geometry and Token-Identity Prediction Optima

For this analysis, let t denote the target token identity of an occurrence, let t' denote a fixed target token identity, and let c index a supervised prediction context whose next-token target is t. The error head receives the corresponding hidden representation $\mathbf{h}_{t,c}$ at the prediction position. Together with the selected native-affinity weights, its expert scores form the positive aggregate $\widehat{e}_{t,c}$ in Equation 4. Let $\mathcal{L}_{\mathrm{CE},t,c}$ denote the realized next-token loss for that occurrence. We omit the layer index and initially treat each prediction as an independently adjustable scalar while holding the observed losses fixed. The pair (t, c) identifies one occurrence; elsewhere, the main text uses t alone as a token-position index. Expectations below are empirical averages or population expectations with finite mean token loss.

Scalar loss geometry. For one contextual occurrence, differentiation with respect to the aggregate prediction gives

$$\frac{\partial \mathcal{L}_{\mathrm{IS},t,c}}{\partial \widehat{e}_{t,c}} = \frac{\widehat{e}_{t,c} - \mathcal{L}^{+}_{\mathrm{CE},t,c}}{\widehat{e}^{2}_{t,c}}, \tag{16}$$

$$\frac{\partial \mathcal{L}_{\mathrm{ENLL},t,c}}{\partial \widehat{e}_{t,c}} = \frac{\widehat{e}_{t,c} - \mathcal{L}_{\mathrm{CE},t,c}}{\widehat{e}^{2}_{t,c}}. \tag{17}$$

For a positive target, the derivative is negative below the target and positive above it. Thus, an independently adjustable occurrence-level prediction is minimized at its realized target. If an ENLL target is zero, its objective is $\log \widehat{e}_{t,c}$ and tends to $-\infty$ as $\widehat{e}_{t,c} \to 0$; the ENLL statements below assume positive targets for the occurrence-level optimum. A positive mean target suffices for the context-averaged ENLL optimum below.

Token-identity optimum. For a fixed target token identity t', consider a positive scalar reference $\widehat{e}_{t'}$, held constant inside the context average. This reference summarizes the identity's supervision target; it is not defined as the average output of the implemented contextual head. Its objectives are

$$\overline{\mathcal{L}}_{\mathrm{IS},t'}(\widehat{e}_{t'}) := \mathbb{E}_{c|t=t'}\left[\frac{\mathcal{L}^{+}_{\mathrm{CE},t,c}}{\widehat{e}_{t'}} - \log\left(\frac{\mathcal{L}^{+}_{\mathrm{CE},t,c}}{\widehat{e}_{t'}}\right) - 1\right], \tag{18}$$

$$\overline{\mathcal{L}}_{\mathrm{ENLL},t'}(\widehat{e}_{t'}) := \mathbb{E}_{c|t=t'}\left[\frac{\mathcal{L}_{\mathrm{CE},t,c}}{\widehat{e}_{t'}} + \log \widehat{e}_{t'}\right]. \tag{19}$$

The overline denotes averaging over contexts whose target token identity is t'. In a finite corpus, this expectation is the arithmetic mean over those occurrences, so frequent contexts contribute according to their empirical frequency. Differentiation yields

$$\frac{\partial \overline{\mathcal{L}}_{\mathrm{IS},t'}}{\partial \widehat{e}_{t'}} = \frac{\widehat{e}_{t'} - \mathbb{E}_{c|t=t'}[\mathcal{L}^{+}_{\mathrm{CE},t,c}]}{\widehat{e}^{2}_{t'}}, \tag{20}$$

$$\frac{\partial \overline{\mathcal{L}}_{\mathrm{ENLL},t'}}{\partial \widehat{e}_{t'}} = \frac{\widehat{e}_{t'} - \mathbb{E}_{c|t=t'}[\mathcal{L}_{\mathrm{CE},t,c}]}{\widehat{e}^{2}_{t'}}. \tag{21}$$

Therefore, the unique positive token-identity optima are

$$\widehat{e}^{\star}_{\mathrm{IS},t'} = \mathbb{E}_{c|t=t'}[\mathcal{L}^{+}_{\mathrm{CE},t,c}], \qquad \widehat{e}^{\star}_{\mathrm{ENLL},t'} = \mathbb{E}_{c|t=t'}[\mathcal{L}_{\mathrm{CE},t,c}]. \tag{22}$$

The derivatives change sign from negative to positive at these values, proving global optimality on the positive scalar domain. The IS floor guarantees a positive IS optimum; ENLL requires a positive conditional mean. This establishes Equation 10 as a token-identity scalar reference. Away from the numerical IS floor, IS and ENLL have the same reference target.

Shared-head interpretation. The implemented prediction $\widehat{e}_{t,c}$ can vary across contexts through $\mathbf{h}_{t,c}$, the affinity weights, and the active expert set. Its direct parameter gradient averages the scalar derivatives above multiplied by the corresponding prediction gradients. Parameter sharing therefore

does not guarantee either occurrence-level loss matching or equality between the average prediction and average loss within each target identity. In particular, averaging contextual predictions before applying a loss is a different objective from averaging their individual supervision losses. For an unrestricted predictor, the corresponding expected-loss optimum conditions on the information available to that predictor; the next-token target identity is not itself an input to the error head. Equation 10 is consequently a reference for grouped evaluation, not a calibration guarantee for the trained head. During joint training, routing and model updates also change the observed losses; Appendices A.2 and A.3 analyze these gradient paths.

A.2 Gradient Allocation and Expert-Score Identifiability

We now return to the main-text convention in which t indexes a supervised token position. Equation 4 supervises one aggregate prediction even though affinity attenuation uses all N expert-indexed scores and the aggregate reads only the K executed experts. We first characterize how the aggregate objective distributes its direct gradient and then clarify which properties of the individual scores this supervision can identify.

Holding the active set, normalized affinity weights, and observed loss fixed,

$$\frac{\partial \widehat{e}_t}{\partial \widehat{e}_{t,i}} = \bar{p}_{t,i}, \qquad i \in \mathcal{S}_t^{\mathrm{act}}. \tag{23}$$

Applying the chain rule to the two token-error objectives gives

$$\frac{\partial \mathcal{L}_{\mathrm{IS},t}}{\partial \widehat{e}_{t,i}} = \bar{p}_{t,i} \frac{\widehat{e}_t - \mathcal{L}^{+}_{\mathrm{CE},t}}{\widehat{e}_t^2}, \tag{24}$$

$$\frac{\partial \mathcal{L}_{\mathrm{ENLL},t}}{\partial \widehat{e}_{t,i}} = \bar{p}_{t,i} \frac{\widehat{e}_t - \mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t^2}. \tag{25}$$

Because $\bar{p}_{t,i} \geq 0$, every active expert score receives a gradient with the same sign as the aggregate prediction gradient. Its magnitude is scaled by $\bar{p}_{t,i}$: among experts receiving the same aggregate error signal, experts with larger native affinity receive proportionally larger direct score gradients. Parameter updates also depend on the head's Jacobian and the optimizer.

This gradient allocation does not make the individual scores independently identifiable from one fixed aggregate. For perturbations $\Delta_{t,i}$ small enough to preserve positive scores and the active set,

$$\sum_{i \in \mathcal{S}_t^{\mathrm{act}}} \bar{p}_{t,i} \Delta_{t,i} = 0 \quad \Longrightarrow \quad \sum_{i \in \mathcal{S}_t^{\mathrm{act}}} \bar{p}_{t,i} (\widehat{e}_{t,i} + \Delta_{t,i}) = \widehat{e}_t. \tag{26}$$

For $K > 1$, this single scalar constraint leaves a local $(K-1)$-dimensional family of active-score perturbations. This is an occurrence-level property of the readout, not a proof of parameter non-identifiability across the dataset. Such perturbations can change attenuated aggregation weights and the resulting CE even when the expert set stays fixed, so they need not preserve the complete training loss. The aggregate target alone does not identify $\widehat{e}_{t,i}$ as the counterfactual loss obtained by executing expert i alone.

Differences among expert-indexed scores can nevertheless emerge because the error head is shared across tokens with different representations, affinities, and active sets. We therefore interpret their relative values as learned error-aware routing signals and evaluate them through affinity attenuation, without claiming independently calibrated expert losses. The displayed derivatives isolate the direct path through the aggregate readout; full parameter gradients may additionally pass through $\bar{p}_{t,i}$, $\mathbf{h}_t$, and the realized CE via attenuation. Unselected scores have zero direct readout gradient but may receive gradients through the routed model where its normalization policy permits. All local derivatives hold away from changes in the discrete top-K set.

A.3 Token-Loss Gradient Analysis

For clarity, we omit the numerical IS floor and take $\mathcal{L}_{\mathrm{CE},t} > 0$. Although IS and ENLL have the same optimum for $\widehat{e}_t$, their derivatives with respect to the observed token loss differ. For IS,

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathrm{IS},t}}{\partial \mathcal{L}_{\mathrm{CE},t}} &= \frac{\partial}{\partial \mathcal{L}_{\mathrm{CE},t}}\left[\frac{\mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t} - \log\left(\frac{\mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t}\right) - 1\right] \\ &= \frac{1}{\widehat{e}_t} - \frac{1}{\mathcal{L}_{\mathrm{CE},t}}. \end{aligned} \tag{27}$$

For ENLL,

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathrm{ENLL},t}}{\partial \mathcal{L}_{\mathrm{CE},t}} &= \frac{\partial}{\partial \mathcal{L}_{\mathrm{CE},t}}\left[\log \widehat{e}_t + \frac{\mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t}\right] \\ &= \frac{1}{\widehat{e}_t}. \end{aligned} \tag{28}$$

Let θ denote any trainable parameters of the routed model, including the error head when applicable. Away from top-K boundaries, applying the chain rule to the complete auxiliary objectives gives

$$\nabla_\theta \mathcal{L}_{\mathrm{IS},t} = \underbrace{\left(\frac{1}{\widehat{e}_t} - \frac{1}{\mathcal{L}_{\mathrm{CE},t}}\right)\nabla_\theta \mathcal{L}_{\mathrm{CE},t}}_{\text{gradient through the token loss}} + \underbrace{\frac{\widehat{e}_t - \mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t^2}\nabla_\theta \widehat{e}_t}_{\text{gradient through the error prediction}}, \tag{29}$$

$$\nabla_\theta \mathcal{L}_{\mathrm{ENLL},t} = \underbrace{\frac{1}{\widehat{e}_t}\nabla_\theta \mathcal{L}_{\mathrm{CE},t}}_{\text{gradient through the token loss}} + \underbrace{\frac{\widehat{e}_t - \mathcal{L}_{\mathrm{CE},t}}{\widehat{e}_t^2}\nabla_\theta \widehat{e}_t}_{\text{gradient through the error prediction}}. \tag{30}$$

The second term is identical for IS and ENLL and trains the aggregate prediction toward the observed token loss. The difference lies in the first term, which changes how each auxiliary objective contributes to the model's task-loss gradient. At the matched prediction $\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t}$, the auxiliary gradients reduce to

$$\nabla_\theta \mathcal{L}_{\mathrm{IS},t}\big|_{\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t}} = 0, \qquad \nabla_\theta \mathcal{L}_{\mathrm{ENLL},t}\big|_{\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t}} = \frac{1}{\mathcal{L}_{\mathrm{CE},t}}\nabla_\theta \mathcal{L}_{\mathrm{CE},t}. \tag{31}$$

For a next-token CE primary objective, we isolate the corresponding per-token terms from Equation 15, suppressing token and layer averages and unrelated auxiliary losses. These combined-gradient identities concern that objective, not the answer-choice CE task term used in our MCQA experiments; the auxiliary-gradient identities above still apply. At the matched prediction,

$$\begin{aligned}\nabla_\theta\left(\mathcal{L}_{\mathrm{CE},t} + \lambda_m \mathcal{L}_{\mathrm{IS},t}\right)\big|_{\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t}} &= \nabla_\theta \mathcal{L}_{\mathrm{CE},t}, \\ \nabla_\theta\left(\mathcal{L}_{\mathrm{CE},t} + \lambda_m \mathcal{L}_{\mathrm{ENLL},t}\right)\big|_{\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t}} &= \left(1 + \frac{\lambda_m}{\mathcal{L}_{\mathrm{CE},t}}\right)\nabla_\theta \mathcal{L}_{\mathrm{CE},t}. \end{aligned} \tag{32}$$

These identities require pointwise equality to the realized loss, not merely the context-averaged reference optimum in Equation 10. At pointwise equality, IS adds no gradient to CE for the current error-aware routed model; this does not imply equality to the native-router baseline update. ENLL still amplifies that model's CE gradient by the displayed inverse-loss factor. The factor alone does not determine absolute gradient magnitudes across tokens. For multiple supervised layers, the same conclusion holds if every layer's prediction matches the token loss; otherwise the layer contributions must be averaged as in Equation 14.

Away from equality, the IS coefficient on the CE path is $1 + \lambda_m(1/\widehat{e}_t - 1/\mathcal{L}_{\mathrm{CE},t})$, which can be negative; the prediction-gradient term must also be included. Thus, the analysis does not prove that every joint update decreases CE. Below the IS floor the IS gradient through the observed loss is zero, while its prediction gradient uses $\mathcal{L}^+_{\mathrm{CE},t}$; the floor boundary requires a subgradient convention. Finally, along the idealized path $\widehat{e}_t = \mathcal{L}_{\mathrm{CE},t} \to 0$, ENLL equals $1 + \log \mathcal{L}_{\mathrm{CE},t}$ and is unbounded below. This is a property of the unconstrained joint objective, not evidence that a finite training run attains that limit.

A.4 STARTED-LOG ROUTING GEOMETRY

This section derives the probability-space form of the attenuation mechanism and characterizes how predicted error changes expert ranking. From Equation 5, exponentiating the adjusted logit gives

$$\exp(\widetilde{a}_{t,i}) = \exp(a_{t,i}) \left(1 + \frac{\widehat{e}_{t,i}}{\tau}\right)^{-\gamma}. \tag{33}$$

Using the native affinity definition in Equation 1 and cancelling its common softmax normalizer yields

$$\begin{aligned}
\widetilde{p}_{t,i} &= \frac{\exp(\widetilde{a}_{t,i})}{\sum_{j=1}^{N} \exp(\widetilde{a}_{t,j})} \\
&= \frac{\exp(a_{t,i}) \left(1 + \widehat{e}_{t,i}/\tau\right)^{-\gamma}}{\sum_{j=1}^{N} \exp(a_{t,j}) \left(1 + \widehat{e}_{t,j}/\tau\right)^{-\gamma}} \\
&= \frac{p_{t,i} \left(1 + \widehat{e}_{t,i}/\tau\right)^{-\gamma}}{\sum_{j=1}^{N} p_{t,j} \left(1 + \widehat{e}_{t,j}/\tau\right)^{-\gamma}}.
\end{aligned} \tag{34}$$

This establishes Equation 7.

For two experts, the common normalizer cancels from their routing odds:

$$\frac{\widetilde{p}_{t,i}}{\widetilde{p}_{t,j}} = \frac{p_{t,i}}{p_{t,j}} \left(\frac{\tau + \widehat{e}_{t,i}}{\tau + \widehat{e}_{t,j}}\right)^{-\gamma}. \tag{35}$$

For $\gamma > 0$, if $\widehat{e}_{t,i} > \widehat{e}_{t,j}$, attenuation reduces the routing odds of expert i relative to expert j. If their predicted errors are equal, the common factor cancels and their native affinity ratio is preserved. The method therefore changes expert ranking through relative differences in predicted error rather than through a uniform shift shared by all experts. Individual normalized probabilities need not all decrease: their changes also depend on the common normalizer. At $\gamma = 0$, native probabilities are recovered.

The reference scale τ makes the transformation sensitive to the magnitude of predicted error relative to a fixed operating scale. For any multiplicative rescaling $\xi > 0$,

$$\phi_\tau(\xi \widehat{e}_{t,i}) = \phi_{\tau/\xi}(\widehat{e}_{t,i}). \tag{36}$$

Thus, rescaling all error predictions while holding τ fixed can change the route. This differs from pure logarithmic attenuation, for which a common multiplicative rescaling contributes only a shared logit shift that cancels under softmax.

The local sensitivities with respect to predicted error and log-error are

$$\frac{\partial \widetilde{a}_{t,i}}{\partial \widehat{e}_{t,i}} = -\frac{\gamma}{\tau + \widehat{e}_{t,i}}, \qquad \frac{\partial \widetilde{a}_{t,i}}{\partial \log \widehat{e}_{t,i}} = -\gamma \frac{\widehat{e}_{t,i}}{\tau + \widehat{e}_{t,i}}. \tag{37}$$

Sensitivity to predicted error is therefore bounded by γ/τ near zero, while sensitivity in log-error coordinates increases smoothly toward γ.

The two asymptotic regimes make the role of τ explicit:

$$\phi_\tau(\widehat{e}_{t,i}) = \begin{cases} \dfrac{\widehat{e}_{t,i}}{\tau} + \mathcal{O}\left(\dfrac{\widehat{e}_{t,i}^2}{\tau^2}\right), & \widehat{e}_{t,i}/\tau \to 0, \\ \log \dfrac{\widehat{e}_{t,i}}{\tau} + \mathcal{O}\left(\dfrac{\tau}{\widehat{e}_{t,i}}\right), & \widehat{e}_{t,i}/\tau \to \infty. \end{cases} \tag{38}$$

At low error, attenuation is approximately linear and bounded in sensitivity. At high error, it recovers logarithmic relative-error routing up to a common shift.

The reference scale τ and attenuation strength γ have different roles: τ sets the transition between the linear and logarithmic regimes, whereas γ scales the overall routing adjustment. We set both to one in the default configuration. This routing transformation does not change the direct IS or ENLL scalar prediction optimum for fixed observed losses derived in Appendix A.1; it changes how the learned prediction affects expert selection and can therefore change joint training dynamics.

A.5 Error-to-Routing Transformation Ablation

The started-log route in Equation 5 uses the absolute operating range of the error head relative to τ. We compare it with the scale-invariant pure-log control

$$\widetilde{a}_{t,i}^{\log} = a_{t,i} - \gamma \log \widehat{e}_{t,i}. \tag{39}$$

The pure logarithm applies the same routing adjustment to a fixed error ratio at every absolute error scale. The started logarithm instead contracts the low-error region and removes the singular routing sensitivity near zero. Appendix A.4 derives its pairwise form, probability interpretation, and gradients.

Table 4 compares the two transforms on ARC-Challenge with Granite, IS supervision, $\gamma = 1$, error-head learning rate 10^{-5}, and the fixed epoch-3 checkpoint. Correlations are calculated on the test population after averaging each layer–token-identity group over seeds 42–44.

Table 4: Error-to-routing transformation ablation on ARC-Challenge with Granite and IS supervision. Accuracy, NLL, and pointwise Pearson r are epoch-3 means over seeds 42–44. Group-level Pearson r and Spearman ρ are computed after averaging each layer–token-identity group across the same seeds. Bold marks the better observed value in each column and does not imply statistical significance.

Routing transform	Pointwise r	Group r	Group ρ	Test NLL	Accuracy (%)
$\log \widehat{e}$	0.1586	0.3975	0.4977	0.9723	**67.27**
$\log(1 + \widehat{e})$	**0.2486**	**0.4521**	**0.5637**	**0.9673**	66.87

Figure 3 visualizes the same comparison. For each transform, we average every layer–token-identity group over seeds 42–44 and form equal-count deciles ordered by predicted error. Predicted and observed group means are standardized using the corresponding training-population moments, which are then applied unchanged to the test population. This removes the large difference in raw prediction scale while preserving each transform's ordering and out-of-sample shift.

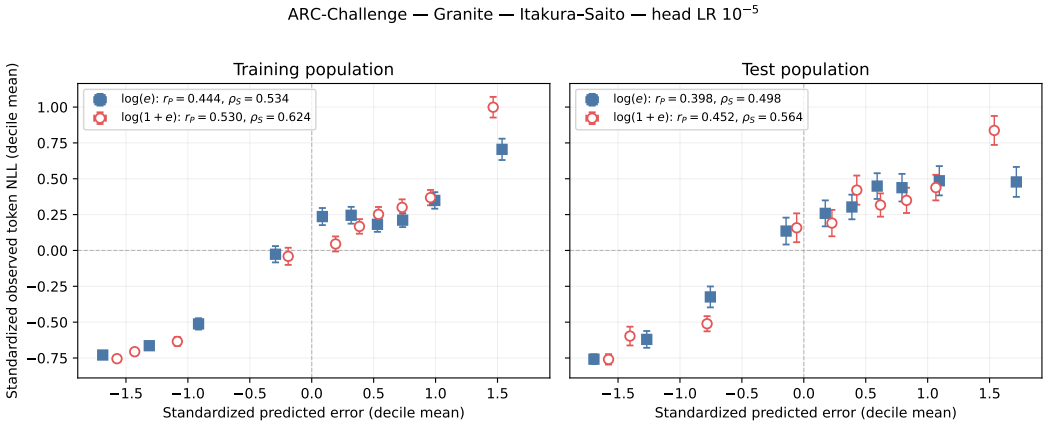


Figure 3: Standardized aggregate error prediction for the pure-log control and the started-log route on ARC-Challenge with Granite, IS supervision, error-head learning rate 10^{-5}, and the fixed epoch-3 checkpoint. Points are equal-count decile means of seed-averaged layer–token-identity groups; whiskers show standard errors across groups within each decile. Legend values report Pearson r_P and Spearman ρ_S over all groups, not over the ten displayed points. Training-population standardization is applied unchanged to test.

Started-log routing increases group-level test Pearson correlation from 0.398 to 0.452 and Spearman correlation from 0.498 to 0.564. The corresponding training correlations increase from 0.444 to 0.530 and from 0.534 to 0.624, respectively. It also improves pointwise correlation and test NLL,

while the pure-log control has 0.40 percentage points higher answer-choice accuracy. The comparison therefore supports a conditioning interpretation: the started-log transform permits a wider raw prediction range while compressing its contribution to routing, and the resulting head ranks token difficulty more consistently. The different accuracy ordering shows that error estimation and useful expert selection remain distinct requirements.

The compared checkpoints also use different neutral error-head initializations: the pure-log route starts at $\widehat{e} = 1$, whereas the started-log route uses the zero-parameter softplus initialization. The observed contrast therefore combines transformation and initialization effects and should not be interpreted as a fully isolated causal estimate.

A.6 TES Hyperparameter Sensitivity

TES introduces two method-specific optimization choices: the learning rate of the token-error head and the attenuation scale γ. We evaluate their sensitivity on Granite ARC-Challenge using started-log IS supervision while keeping the backbone, data, training schedule, and seed cohort fixed.

Table 5: TES hyperparameter sensitivity on Granite ARC-Challenge with started-log IS supervision. Entries are epoch-3 means over seeds 42–44. The learning-rate panel fixes $\gamma = 1$; the attenuation panel fixes the error-head learning rate at 10^{-5}. Bold marks the best observed value within each panel and metric.

Parameter	Value	Accuracy (%)	Choice NLL
Error-head learning rate	10^{-3}	66.07	0.9811
	10^{-4}	66.60	0.9691
	10^{-5}	**66.87**	**0.9673**
	5×10^{-6}	66.67	**0.9673**
Attenuation scale γ	0.5	65.87	0.9729
	1	**66.87**	0.9673
	2	66.00	**0.9654**

Table 5 shows that the selected defaults—error-head learning rate 10^{-5} and $\gamma = 1$—give the highest observed accuracy in the corresponding sweeps. Nearby settings produce similar results, while the γ sweep also shows that minimizing choice NLL and maximizing answer-choice accuracy need not select the same value. These experiments cover IS on Granite ARC-Challenge.

A.7 Experimental Configuration

Table 6 reports the complete configuration used for the primary downstream experiments. Within each backbone and dataset, all methods use the same registered splits, prompt, candidate scoring rule, adaptation surface, and evaluation procedure.

The registered manifests define custom partitions rather than subsamples of official test sets. The split builder pools the declared source partitions, filters invalid examples, and assigns examples deterministically with seed 42. Source identifiers are retained, with disjoint IDs across the training, validation, and test manifests. The released split metadata record the source partitions and their allocation; examples from an official test partition can therefore occur in the custom training partition. These results should not be interpreted as official-test evaluations.

Our evaluator uses the `manifest_answer_text_choice_ce` profile. The prompt is `Question: <question>` followed by a newline and `Answer:`; each candidate appends one space followed by its answer text, without a chat wrapper or in-context demonstrations. Candidates are ranked by mean conditional token log-probability. Accuracy is the fraction of correctly selected answers, denoted `acc_norm` in the evaluation artifacts. The primary adaptation objective is CE over these answer-choice scores, whereas TES and ACS use observed next-token CE as their auxiliary target. Evaluation uses the repository's manifest-based candidate scorer rather than an unmodified benchmark-harness invocation.

Table 6: Backbone, optimization, and method configurations for the primary downstream evaluation. ARC denotes ARC-Challenge; "others" denotes OpenBookQA, SciQ, and MedMCQA.

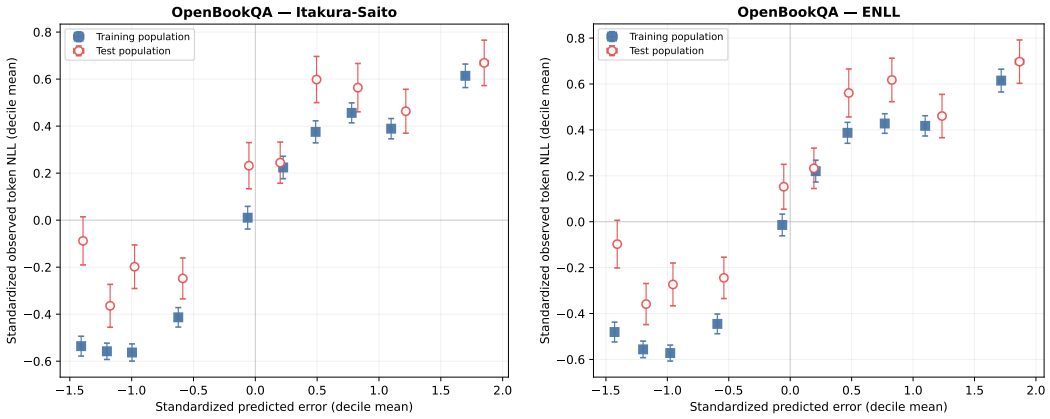
Setting	Granite 3.1 3B-A800M	OLMoE-1B-7B-SFT
Training examples	2,000 (ARC); 4,992 (others)	2,000 (ARC); 5,000 (others)
Validation / test examples	50 / 500	50 / 500
LoRA rank / α / dropout	8 / 8 / 0.05	8 / 8 / 0.05
LoRA targets	Q/K/V and routed-expert projections	Q/K/V/O and routed-expert projections
LoRA learning rate	10^{-4}	2×10^{-5}
Batch / accumulation	8 / 2	8 / 1
Optimizer	AdamW	AdamW
Schedule / warmup	cosine / 0.05	linear / 0.03
Weight decay / gradient clip	0 / 1	0 / 1
Precision	BF16	BF16
IS observation floor $\varepsilon_{\mathrm{IS}}$	10^{-8}	10^{-8}
Steps per epoch	125 (ARC); 312 (others)	250 (ARC); 625 (others)
Epochs / seeds	3 / 42, 43, 45	3 / 42, 43, 45
Supervised MoE scope	final sparse layer	final sparse layer
Native router / auxiliary loss	frozen / 0	frozen / 0
Method	Objective and method-specific configuration	Evaluation route
CE	task CE only	native affinity
Dual Affinity	task CE; copied-head LR 10^{-3}; mixing weight 0.5	Dual Affinity
TES–IS	task CE $+ 10^{-3}\mathcal{L}_{\mathrm{IS}}$; error-head LR 10^{-5}	started-log attenuation, $\gamma = \tau = 1$
TES–ENLL	task CE $+ 10^{-3}\mathcal{L}_{\mathrm{ENLL}}$; error-head LR 10^{-5}	started-log attenuation, $\gamma = \tau = 1$
ACS–IS	task CE $+ 10^{-3}\mathcal{L}_{\text{ACS-IS}}$; no method head	native affinity
ACS–ENLL	task CE $+ 10^{-3}\mathcal{L}_{\text{ACS-ENLL}}$; no method head	native affinity

Every epoch checkpoint is saved. Epoch 3 is the predeclared primary endpoint. For the secondary robustness analysis, one checkpoint is selected independently for each seed by minimum length-normalized answer-choice NLL on the 50-example validation split; exact ties select the earlier epoch. The selection is fixed before test evaluation, and test metrics do not enter the selection rule.

A.8 Cross-Dataset Error-Prediction Analysis

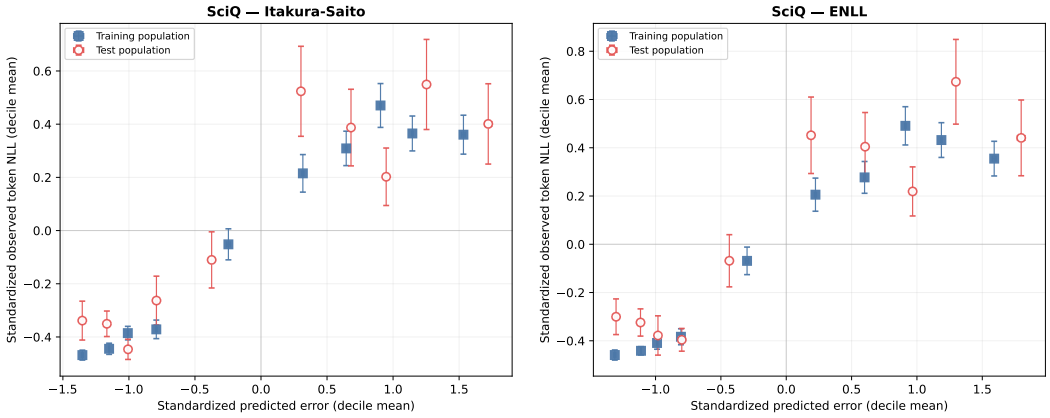
Figures 4– 6 extend the ARC-Challenge analysis in Figure 2 to the remaining datasets using the Granite $\log(1+\widehat{e})$ experiments. For each objective and dataset, we use the epoch-3 checkpoint with $\gamma = 1$, error-head learning rate 10^{-5}. Within each seed, we average predicted and observed errors over the contextual occurrences in each layer–token-identity group. We then average group means across seeds and form equal-count deciles ordered by predicted error. Predicted and observed group means are standardized by their respective training-population moments, which are then applied unchanged to the test population. Each panel uses automatically selected axis limits for visibility. The analysis evaluates the aggregate prediction; it does not interpret the expert-indexed scores as observed counterfactual expert errors.

Both OpenBookQA objectives retain positive out-of-sample ordering. Across test groups, Pearson/Spearman correlations are $0.316/0.333$ for IS and $0.321/0.344$ for ENLL. The highest predicted-error decile also has a larger standardized observed NLL than the lowest decile for both objectives. Because the axes are standardized separately from the training reference, this figure assesses association and distribution shift rather than absolute NLL calibration.



(a) Itakura–Saito **(b) ENLL**

Figure 4: Standardized aggregate error prediction on OpenBookQA for the Granite $\log(1+\widehat{e})$ runs. Points are equal-count decile means after averaging layer–token-identity groups over seeds 42–44; whiskers show ± 1 standard error across groups. Blue squares and red circles denote the training and test populations. Axis limits are selected automatically for each panel.



(a) Itakura–Saito **(b) ENLL**

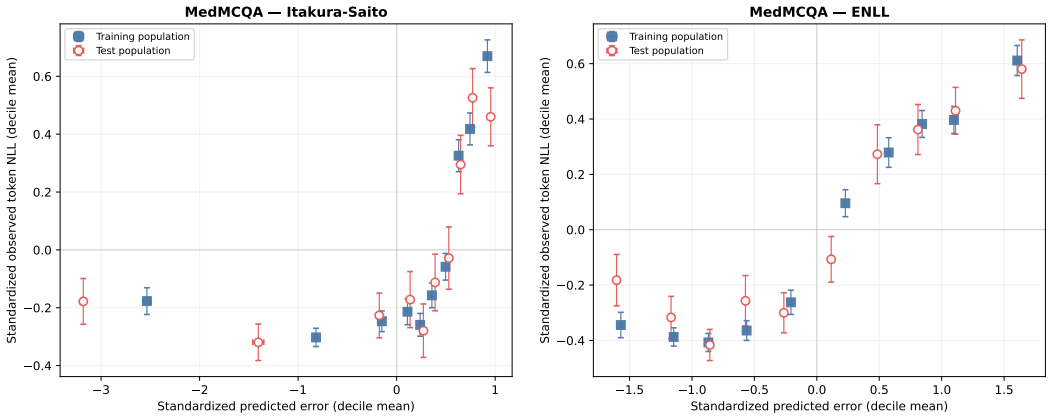
Figure 5: Standardized aggregate error prediction on SciQ for the Granite $\log(1+\widehat{e})$ runs. Plot construction and visual encoding follow Figure 4.

SciQ also shows positive test-set group association, with Pearson/Spearman correlations of $0.300/0.378$ for IS and $0.308/0.377$ for ENLL. The decile trajectories are not monotone at every adjacent transition, but both objectives separate their lowest and highest predicted-error deciles in the expected direction.

MedMCQA ENLL retains positive test-set group association (Pearson/Spearman $0.297/0.353$) and a positive endpoint separation. The IS aggregate is weaker ($0.142/0.240$). Across the evaluated datasets, the normalized plots show that both objectives learn a positive token-error ordering, while the strength of that ordering remains dataset dependent.

A.9 Affinity-Concentration Supervision

Affinity-concentration supervision (ACS) is a headless alternative to the learned error-head method. It uses token-loss supervision to shape the native router distribution directly. ACS adds no error-specific parameters and applies no error-dependent transformation at inference. Expert selection and aggregation therefore continue to follow the native route in Equations 1– 2.



(a) Itakura–Saito **(b) ENLL**

Figure 6: Standardized aggregate error prediction on MedMCQA for the Granite $\log(1+\widehat{e})$ runs. Plot construction and visual encoding follow Figure 4.

For supervised token position t in layer l, let $\mathcal{S}_t^{(l)}$ be the native top-K expert set. We normalize the native affinity probabilities within this set and measure their squared concentration:

$$\bar{p}_{t,i}^{(l)} = \frac{p_{t,i}^{(l)}}{\sum_{j\in\mathcal{S}_t^{(l)}} p_{t,j}^{(l)}}, \qquad C_t^{(l)} = \sum_{i\in\mathcal{S}_t^{(l)}} \left(\bar{p}_{t,i}^{(l)}\right)^2. \tag{40}$$

The concentration satisfies $1/K \le C_t^{(l)} \le 1$: Cauchy–Schwarz gives $1 = (\sum_i \bar{p}_{t,i}^{(l)})^2 \le K C_t^{(l)}$, and $\sum_i (\bar{p}_{t,i}^{(l)})^2 \le (\sum_i \bar{p}_{t,i}^{(l)})^2 = 1$, where both sums are over the selected set. The lower endpoint corresponds to equal selected affinities. For $K > 1$ and finite softmax logits, all selected probabilities are positive, so $C_t^{(l)} < 1$; the upper endpoint is approached as affinity concentrates on one expert. For $K = 1$, concentration is identically one and supplies no direct concentration gradient. Equivalently, $1/C_t^{(l)}$ is the effective number of selected experts and $-\log C_t^{(l)}$ is their order-two Rényi entropy (Rényi, 1961; Hill, 1973). Thus, $C_t^{(l)}$ is a bounded description of router sharpness, not a calibrated estimate of token or expert error.

For evaluation, we compute Equation 40 at every valid supervised token position using the exact expert set executed by the native forward pass. Let $\mathcal{T}_{\mathcal{D}}$ denote these token positions for evaluation split $\mathcal{D}$. We average over these positions within each checkpoint, then report the mean and sample standard deviation across seeds, weighting each seed equally. Appendix A.12 defines the reported statistics. When multiple layers are supervised, concentration is reported separately by layer unless a cross-layer average is explicitly identified. Within-token affinity concentration is distinct from expert-load concentration across a dataset.

ACS applies the same IS or ENLL form used by the error-head method, but substitutes the native concentration for the learned aggregate prediction:

$$\mathcal{L}_{\text{ACS-IS},t}^{(l)} = \frac{\mathcal{L}_{\text{CE},t}^{+}}{C_t^{(l)}} - \log\left(\frac{\mathcal{L}_{\text{CE},t}^{+}}{C_t^{(l)}}\right) - 1, \tag{41}$$

$$\mathcal{L}_{\text{ACS-ENLL},t}^{(l)} = \log C_t^{(l)} + \frac{\mathcal{L}_{\text{CE},t}}{C_t^{(l)}}. \tag{42}$$

The two objectives have the direct concentration gradients

$$\frac{\partial \mathcal{L}^{(l)}_{\text{ACS-IS},t}}{\partial C^{(l)}_t} = \frac{C^{(l)}_t - \mathcal{L}^{+}_{\text{CE},t}}{\left(C^{(l)}_t\right)^2}, \tag{43}$$

$$\frac{\partial \mathcal{L}^{(l)}_{\text{ACS-ENLL},t}}{\partial C^{(l)}_t} = \frac{C^{(l)}_t - \mathcal{L}_{\text{CE},t}}{\left(C^{(l)}_t\right)^2}. \tag{44}$$

When the IS floor is inactive, these gradients coincide. With the selected set fixed, its normalization cancels the all-expert softmax denominator: $\bar{p}^{(l)}_{t,i} = \exp(a^{(l)}_{t,i}) / \sum_{j \in \mathcal{S}^{(l)}_t} \exp(a^{(l)}_{t,j})$. Consequently, differentiating the sum of squared selected probabilities gives

$$\frac{\partial C^{(l)}_t}{\partial a^{(l)}_{t,i}} = 2\bar{p}^{(l)}_{t,i}\left(\bar{p}^{(l)}_{t,i} - C^{(l)}_t\right), \qquad i \in \mathcal{S}^{(l)}_t. \tag{45}$$

For $i \notin \mathcal{S}^{(l)}_t$, the local derivative is zero while the selected set remains fixed; top-K boundaries are nondifferentiable. Multiplying this derivative by the corresponding concentration derivative above gives the direct auxiliary gradient for a selected logit, holding CE fixed. For this path alone, gradient descent favors greater concentration when the target exceeds $C^{(l)}_t$, and less concentration when it is smaller. At exactly uniform selected affinity, however, $\bar{p}^{(l)}_{t,i} = C^{(l)}_t = 1/K$ and every concentration-logit derivative is zero. Near this point, and near single-expert saturation, the direct gradient can be small. Thus, the scalar gradient's sign does not guarantee a concentration change in a parameter update, which also includes CE and shared-parameter effects. ACS does not identify which unselected expert would reduce the loss and supplies no counterfactual expert-error target.

The bounded range also prevents ACS from inheriting the unrestricted error-prediction interpretation of Section 3.3. Across the contextual occurrences c of a fixed target token identity t', consider one scalar concentration reference $C^{(l)}_{t'}$ held constant inside the average, as in Appendix A.1. Its expected-objective derivative is $(C^{(l)}_{t'} - \mathbb{E}_{c|t=t'}[\mathcal{L}^{+}_{\text{CE},t,c}])/(C^{(l)}_{t'})^2$ for IS, with the unfloored loss for ENLL. Its sign changes at the mean target, so minimization over the closed interval $[1/K, 1]$ gives

$$C^{(l)\star}_{\text{IS},t'} = \text{clip}\left(\mathbb{E}_{c|t=t'}[\mathcal{L}^{+}_{\text{CE},t,c}], \frac{1}{K}, 1\right), \qquad C^{(l)\star}_{\text{ENLL},t'} = \text{clip}\left(\mathbb{E}_{c|t=t'}[\mathcal{L}_{\text{CE},t,c}], \frac{1}{K}, 1\right). \tag{46}$$

Here clip restricts its first argument to $[1/K, 1]$. These scalar reference optima are not the average of independently optimized contextual concentrations: averaging and clipping do not generally commute. The native router further couples occurrences through shared parameters. For $K > 1$, when the scalar reference optimum is one, softmax routing approaches the objective's infimum as selected logit differences grow; no finite logits attain it. Expected losses at least one therefore share the same upper-boundary reference. Moreover, because the observed token loss is not detached, IS and ENLL retain different gradients through the language model even when their direct concentration gradients agree. For $\mathcal{L}_{\text{CE},t} > \varepsilon_{\text{IS}}$,

$$\frac{\partial \mathcal{L}^{(l)}_{\text{ACS-IS},t}}{\partial \mathcal{L}_{\text{CE},t}} = \frac{1}{C^{(l)}_t} - \frac{1}{\mathcal{L}_{\text{CE},t}}, \qquad \frac{\partial \mathcal{L}^{(l)}_{\text{ACS-ENLL},t}}{\partial \mathcal{L}_{\text{CE},t}} = \frac{1}{C^{(l)}_t}. \tag{47}$$

Below the IS floor, the IS derivative through the observed loss is zero; the ENLL expression is unchanged. The full chain rule adds this CE path to the concentration path, exactly as in Appendix A.3 with $\widehat{e}_t$ replaced by $C^{(l)}_t$.

The ACS training objective follows Equation 15, with $m \in \{\text{ACS-IS}, \text{ACS-ENLL}\}$ and the generic coefficient λ_m. Through trainable parameters that affect the affinity logits, ACS can alter future native routes, but it does not replace the native routing algorithm during either training or evaluation.

Table 1 reports the complete Granite and OLMoE ACS results. The two ACS objectives remain close across datasets, and neither is uniformly preferred. On Granite, ACS-IS attains the highest observed mean on ARC-Challenge and SciQ; the corresponding OLMoE results are competitive but do not exceed learned-head IS on ARC-Challenge or OpenBookQA.

Table 7: ACS coefficient sensitivity (accuracy, %) under final-layer supervision. Bold marks the higher mean within each matched pair.

Model	Affinity	Objective	λ_m	ARC-C	OpenBook QA	SciQ	MedMCQA
Granite	Trainable	IS	10^{-3}	70.60	73.93	**89.53**	40.67
			10^{-2}	**71.20**	**74.53**	88.80	**41.07**
		ENLL	10^{-3}	69.87	73.53	**89.20**	**42.13**
			10^{-2}	**70.47**	**75.20**	89.00	41.47
OLMoE	Frozen	IS	10^{-3}	62.73	**69.93**	88.87	**44.20**
			10^{-2}	**63.40**	69.80	**89.67**	44.07
		ENLL	10^{-3}	62.93	**70.20**	88.73	**44.53**
			10^{-2}	**63.93**	69.20	**90.07**	44.13
OLMoE	Trainable	IS	10^{-3}	63.60	**69.33**	89.07	**44.53**
			10^{-2}	**64.20**	69.27	**89.47**	43.73
		ENLL	10^{-3}	63.80	**71.00**	88.87	**44.27**
			10^{-2}	**64.47**	69.60	**89.80**	43.67

Table 8: Complete frozen-affinity OLMoE ARC-Challenge coefficient screen.

Objective	10^{-4}	10^{-3}	10^{-2}	10^{-1}
ACS–IS	63.27 ± 0.12	62.73 ± 0.31	$\mathbf{63.40 \pm 0.40}$	59.00 ± 0.40
ACS–ENLL	62.33 ± 0.46	62.93 ± 0.64	$\mathbf{63.93 \pm 0.76}$	58.33 ± 1.21

For the frozen-affinity ARC-Challenge epoch-3 runs with final-layer ACS and $\lambda_m = 10^{-3}$, mean concentration remains between 0.1394 and 0.1409, close to the uniform-routing endpoint $1/K = 0.125$ for $K = 8$, while its Pearson correlation with observed token CE is weak and negative (-0.04 to -0.14). These diagnostics do not establish concentration as a token-error predictor or isolate whether the accuracy gains arise from concentration regularization, attached-loss gradient shaping, or both.

A.10 ACS Supervision-Coefficient Sensitivity

The primary comparison fixes $\lambda_m = 10^{-3}$ across datasets. We test the sensitivity of ACS to this choice by comparing it with $\lambda_m = 10^{-2}$ under otherwise matched final-layer configurations. We retain every evaluated dataset in Table 7, including those for which the larger coefficient reduces accuracy.

The larger coefficient improves five of eight Granite trainable-affinity objective–dataset cells and four of eight cells in each OLMoE affinity setting. The largest increases are 1.67 percentage points for Granite ACS–ENLL on OpenBookQA and 1.34 points for frozen-affinity OLMoE ACS–ENLL on SciQ. Table 8 also shows that increasing the coefficient further to 10^{-1} sharply reduces OLMoE ARC-Challenge accuracy. Thus, 10^{-2} is a useful dataset-dependent alternative rather than a uniformly better default.

The OLMoE trainable-affinity comparison uses validation-NLL-selected checkpoints.

A.11 Affinity Supervision Depth Ablation

Section 4.4 isolates supervision depth while keeping native affinity frozen. Here, we study the coupled setting in which ACS supervision and native-affinity training are applied to the same layer scope. We compare the final layer, first half, last half, and full MoE stack. Consequently, differences between scopes reflect both the placement of token-loss supervision and the number of affinity routers being optimized.

Table 9 reports the fixed epoch-3 Granite endpoint at $\lambda_m = 10^{-3}$. For Granite, final-layer training gives the highest ACS accuracy for both objectives on all four datasets, whereas all-layer training is consistently weakest. The matched Router-CE control also degrades as affinity training broadens, showing that the all-layer decline is not specific to ACS. Relative to the matched control at each scope, the clearest additional benefit from ACS occurs on ARC-Challenge under final-layer

Table 9: Answer-choice accuracy (%; mean ± sample standard deviation over seeds 42, 43, and 45) under trainable native-affinity layer scopes. Granite values use the fixed epoch-3 checkpoint and $\lambda_m = 10^{-3}$. OLMoE ACS values marked * use the checkpoint selected by minimum validation NLL; ACS rows use $\lambda_m = 10^{-2}$. Bold marks the best evaluated scope within each model, method, and dataset.

Model	Method	ACS and affinity-training scope	ARC-Challenge	OpenBookQA	SciQ	MedMCQA
Granite 3.1	Router CE	Final 1	$\mathbf{67.53 \pm 1.14}$	$\mathbf{74.00 \pm 0.72}$	88.60 ± 0.40	$\mathbf{41.73 \pm 1.22}$
		First 8	65.60 ± 3.02	70.93 ± 2.61	$\mathbf{89.07 \pm 0.23}$	40.87 ± 0.42
		Last 8	65.80 ± 1.51	71.07 ± 0.61	88.07 ± 0.61	37.73 ± 0.90
		All 32	61.60 ± 4.20	61.60 ± 3.40	86.27 ± 0.42	30.80 ± 3.34
	ACS-ENLL	Final 1	$\mathbf{69.87 \pm 1.33}$	$\mathbf{73.53 \pm 0.90}$	$\mathbf{89.20 \pm 0.92}$	$\mathbf{42.13 \pm 1.27}$
		First 8	65.00 ± 1.04	71.33 ± 1.10	88.87 ± 1.53	40.53 ± 2.00
		Last 8	68.40 ± 1.31	69.60 ± 3.67	88.40 ± 0.80	38.60 ± 0.72
		All 32	63.60 ± 0.92	61.13 ± 9.99	71.40 ± 26.33	35.87 ± 1.10
	ACS-IS	Final 1	$\mathbf{70.60 \pm 1.80}$	$\mathbf{73.93 \pm 0.31}$	$\mathbf{89.53 \pm 0.42}$	$\mathbf{40.67 \pm 1.45}$
		First 8	67.73 ± 1.27	70.00 ± 1.40	88.00 ± 0.69	38.80 ± 1.44
		Last 8	68.47 ± 1.55	70.20 ± 0.35	88.67 ± 1.03	37.40 ± 1.22
		All 32	60.13 ± 10.34	68.87 ± 1.81	87.73 ± 0.81	35.00 ± 2.31
OLMoE	Router CE	Final 1	$63.53 \pm 0.58^*$	$\mathbf{71.87 \pm 0.42}^*$	$\mathbf{88.80 \pm 0.20}^*$	$\mathbf{44.47 \pm 0.46}^*$
		First 8	$63.07 \pm 0.90^*$	$71.20 \pm 0.20^*$	$88.73 \pm 0.23^*$	$43.87 \pm 0.99^*$
		Last 8	$63.53 \pm 0.23^*$	$70.40 \pm 0.53^*$	$\mathbf{88.80 \pm 0.20}^*$	$43.47 \pm 0.64^*$
		All 16	$\mathbf{64.67 \pm 1.40}^*$	$70.53 \pm 0.95^*$	$88.73 \pm 0.31^*$	$43.60 \pm 0.87^*$
	ACS-ENLL	Final 1	$\mathbf{64.47 \pm 0.70}^*$	$69.60 \pm 0.53^*$	$\mathbf{89.80 \pm 0.20}^*$	$43.67 \pm 0.70^*$
		First 8	$\mathbf{64.47 \pm 0.83}^*$	$70.40 \pm 0.20^*$	$88.60 \pm 0.20^*$	$43.87 \pm 1.03^*$
		Last 8	$64.13 \pm 0.31^*$	$69.60 \pm 0.80^*$	$88.67 \pm 0.23^*$	$\mathbf{44.20 \pm 0.40}^*$
		All 16	$63.47 \pm 0.12^*$	$\mathbf{70.80 \pm 0.53}^*$	$88.07 \pm 0.23^*$	$43.13 \pm 1.10^*$
	ACS-IS	Final 1	$64.20 \pm 0.20^*$	$69.27 \pm 0.42^*$	$\mathbf{89.47 \pm 0.42}^*$	$43.73 \pm 0.58^*$
		First 8	$64.07 \pm 1.29^*$	$69.80 \pm 0.20^*$	$89.33 \pm 0.50^*$	$\mathbf{44.27 \pm 0.70}^*$
		Last 8	$63.73 \pm 0.23^*$	$\mathbf{70.00 \pm 0.87}^*$	$88.93 \pm 0.46^*$	$44.07 \pm 0.50^*$
		All 16	$\mathbf{64.27 \pm 0.90}^*$	$\mathbf{70.00 \pm 0.20}^*$	$88.60 \pm 0.35^*$	$43.40 \pm 0.80^*$

and last-half training. These results support narrow affinity supervision rather than indiscriminate optimization of every router.

The OLMoE panel reports validation-NLL-selected checkpoints at $\lambda_m = 10^{-2}$ together with the matched Router-CE controls. These results do not identify a universal placement: ARC-Challenge and SciQ favor narrow or final-layer supervision, whereas OpenBookQA and MedMCQA favor broader or half-depth scopes depending on the objective. Relative to Router CE, ACS helps on SciQ, is roughly neutral on ARC-Challenge and MedMCQA, and hurts on OpenBookQA. We therefore do not compare this panel directly with the frozen-affinity results in Section 4.4, which uses fixed epoch-3 checkpoints.

A.12 Affinity-Concentration Diagnostics

Accuracy alone does not establish whether ACS changes the concentration it directly supervises. Table 10 reports the mean selected-set concentration $\bar{C}_{\mathcal{D}}$, its reciprocal effective expert count $N_{\mathrm{eff},\mathcal{D}}$, and the token-level Pearson correlation $r_{\mathcal{D}}$ between concentration and realized loss for the final-layer Granite Router-CE and ACS runs on the ARC-Challenge train and test sets. In this comparison, the final native affinity router is trainable in all three methods, and ACS uses $\lambda_m = 10^{-2}$. This differs from the frozen-affinity, $\lambda_m = 10^{-3}$ diagnostics in Appendix A.9.

Table 10: Final-layer affinity-concentration diagnostics for Granite on ARC-Challenge with the native affinity router trained by task CE alone (Router CE) or jointly with ACS at $\lambda_m = 10^{-2}$. Entries are epoch-3 means $\pm$ sample standard deviations over seeds 42, 43, and 45. Each seed contains 14,679 train-token and 3,550 test-token observations.

Method	Split	$\bar{C}_{\mathcal{D}}$	$N_{\mathrm{eff},\mathcal{D}}$	$r_{\mathcal{D}}$
Router CE	Train	0.2614 ± 0.0279	3.86 ± 0.43	-0.152 ± 0.083
	Test	0.2589 ± 0.0319	3.90 ± 0.50	-0.161 ± 0.070
ACS–IS	Train	0.8068 ± 0.0395	1.24 ± 0.06	0.145 ± 0.075
	Test	0.8077 ± 0.0397	1.24 ± 0.06	0.143 ± 0.082
ACS–ENLL	Train	0.7813 ± 0.0381	1.28 ± 0.06	0.169 ± 0.087
	Test	0.7830 ± 0.0362	1.28 ± 0.06	0.158 ± 0.101

For one checkpoint and split $\mathcal{D} \in \{\mathrm{train}, \mathrm{test}\}$, let $\mathcal{T}_{\mathcal{D}}$ denote the evaluated token positions. All quantities below are computed at the final sparse MoE layer, and we suppress the layer index as in Section 3.4. For position $t \in \mathcal{T}_{\mathcal{D}}$, $\mathcal{S}_t$ is the native top-K expert set and $p_{t,i}$ is the affinity probability assigned to expert i. The selected-set probability $\bar{p}_{t,i}$ and per-token concentration C_t are

$$\bar{p}_{t,i} = \frac{p_{t,i}}{\sum_{j \in \mathcal{S}_t} p_{t,j}}, \qquad C_t = \sum_{i \in \mathcal{S}_t} \bar{p}_{t,i}^2. \tag{48}$$

Equation 48 is the same order-two concentration defined in Equation 11. Granite uses $K = 8$, so $1/8 \le C_t \le 1$. The checkpoint-level mean concentration and its reciprocal effective expert count are

$$\bar{C}_{\mathcal{D}} = \frac{1}{|\mathcal{T}_{\mathcal{D}}|} \sum_{t \in \mathcal{T}_{\mathcal{D}}} C_t, \qquad N_{\mathrm{eff},\mathcal{D}} = \frac{1}{\bar{C}_{\mathcal{D}}}. \tag{49}$$

Thus, $1/K \le \bar{C}_{\mathcal{D}} \le 1$ and $1 \le N_{\mathrm{eff},\mathcal{D}} \le K$: larger $\bar{C}_{\mathcal{D}}$ and smaller $N_{\mathrm{eff},\mathcal{D}}$ indicate that the selected affinity mass is carried by fewer experts. We first average concentration over token positions within each checkpoint as shown in Equation 49 and then take its reciprocal; this differs from averaging $1/C_t$ over positions.

The final column measures the token-level Pearson association between concentration and realized next-token loss:

$$r_{\mathcal{D}} = \frac{\sum_{t \in \mathcal{T}_{\mathcal{D}}} (C_t - \bar{C}_{\mathcal{D}})(\mathcal{L}_{\mathrm{CE},t} - \bar{\mathcal{L}}_{\mathrm{CE},\mathcal{D}})}{\sqrt{\sum_{t \in \mathcal{T}_{\mathcal{D}}} (C_t - \bar{C}_{\mathcal{D}})^2} \sqrt{\sum_{t \in \mathcal{T}_{\mathcal{D}}} (\mathcal{L}_{\mathrm{CE},t} - \bar{\mathcal{L}}_{\mathrm{CE},\mathcal{D}})^2}}, \qquad \bar{\mathcal{L}}_{\mathrm{CE},\mathcal{D}} = \frac{1}{|\mathcal{T}_{\mathcal{D}}|} \sum_{t \in \mathcal{T}_{\mathcal{D}}} \mathcal{L}_{\mathrm{CE},t}. \tag{50}$$

A positive $r_{\mathcal{D}}$ means that higher-loss tokens tend to have more concentrated selected affinity; a negative value indicates the opposite association. For constant concentration or constant loss, the denominator is zero and the correlation is undefined, rather than zero. For each method and split, Table 10 reports the mean and sample standard deviation of $\bar{C}_{\mathcal{D}}$, $N_{\mathrm{eff},\mathcal{D}}$, and $r_{\mathcal{D}}$ across the three training seeds.

Both ACS objectives produce substantially more concentrated selected-affinity distributions than Router CE: mean concentration increases from approximately 0.26 to 0.78–0.81, while the reciprocal effective count decreases from approximately 3.9 to 1.2–1.3. The association with observed token CE also changes from weakly negative under Router CE to weakly positive under both ACS objectives. These shifts are stable between train and test, but are descriptive and do not establish that increased concentration causes the corresponding accuracy differences. The effective count describes within-token weight concentration: all K selected experts still execute, and this statistic does not measure expert load concentration across tokens.

A.13 Coefficient Scaling Across Supervision Depth

Equation 14 averages the ACS objective over the supervised MoE layers. Holding the global coefficient fixed while increasing the number of supervised layers therefore reduces the effective contribution associated with each layer. We test whether increasing the global coefficient with supervision depth recovers the behavior observed under narrower supervision.

Table 11: ARC-Challenge excessive-scaling control (accuracy, %).

Scope	Objective	Depth-scaled	$10\times$ scaled	Change
Last 8	IS	69.27 ± 0.64	57.73 ± 2.32	-11.53
	ENLL	68.80 ± 1.06	56.13 ± 1.22	-12.67
All 32	IS	67.20 ± 3.47	48.60 ± 0.92	-18.60
	ENLL	69.00 ± 1.11	48.13 ± 0.50	-20.87

Table 12: Granite epoch-3 accuracy (%) under fixed and depth-scaled ACS coefficients. Bold marks the higher matched mean.

Dataset	Scope	Objective	Fixed 10^{-3}	Depth-scaled	Change
ARC-C	Last 8	IS	68.47 ± 1.55	$\mathbf{69.27 \pm 0.64}$	$+0.80$
		ENLL	68.40 ± 1.31	$\mathbf{68.80 \pm 1.06}$	$+0.40$
	All 32	IS	60.13 ± 10.34	$\mathbf{67.20 \pm 3.47}$	$+7.07$
		ENLL	63.60 ± 0.92	$\mathbf{69.00 \pm 1.11}$	$+5.40$
OpenBookQA	Last 8	IS	70.20 ± 0.35	$\mathbf{70.27 \pm 0.12}$	$+0.07$
		ENLL	69.60 ± 3.67	$\mathbf{70.20 \pm 0.53}$	$+0.60$
	All 32	IS	$\mathbf{68.87 \pm 1.81}$	59.20 ± 9.53	-9.67
		ENLL	61.13 ± 9.99	$\mathbf{65.00 \pm 1.22}$	$+3.87$
SciQ	Last 8	IS	88.67 ± 1.03	$\mathbf{89.27 \pm 0.50}$	$+0.60$
		ENLL	88.40 ± 0.80	$\mathbf{88.87 \pm 0.42}$	$+0.47$
	All 32	IS	$\mathbf{87.73 \pm 0.81}$	87.53 ± 1.01	-0.20
		ENLL	71.40 ± 26.33	$\mathbf{88.00 \pm 0.92}$	$+16.60$
MedMCQA	Last 8	IS	37.40 ± 1.22	$\mathbf{38.00 \pm 4.78}$	$+0.60$
		ENLL	38.60 ± 0.72	$\mathbf{38.80 \pm 1.22}$	$+0.20$
	All 32	IS	35.00 ± 2.31	$\mathbf{37.27 \pm 0.81}$	$+2.27$
		ENLL	35.87 ± 1.10	$\mathbf{37.40 \pm 2.43}$	$+1.53$

Table 12 shows that restoring per-layer supervision strength improves all four ARC-Challenge arms relative to the fixed global coefficient. Across the other datasets, ten of twelve scope–objective arms are unchanged or improved. These results show that supervision depth and coefficient cannot be interpreted independently. At the same time, the excessive-scaling control in Table 11 shows that a further tenfold increase sharply reduces accuracy, identifying an intermediate operating range rather than a monotonic benefit from stronger supervision.