WHY BETTER EXPERT CHOICES DO NOT ALWAYS MAKE BETTER MOE ROUTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse mixture-of-experts (MoE) routing determines which experts contribute to each next-token prediction in language models. A router selects a small set of experts from the available prefix before the next token is observed. Yet a search conducted after observing that token can find expert choices that predict it much better than the native route, leaving unclear how much of this gain a router could achieve. Recent methods use such hindsight choices to supervise router training. Because the search sees the answer that the router cannot, part of its gain may be unavailable to any input-only decision. Conversely, a trained router's failure to match the search cannot show whether it has also missed useful choices that were available before the answer. We separate these effects within fixed candidate sets by using independent human completions of the same prefix to lower-bound the value of answer information without assuming an optimal learner. We then use a frozen reference model's complete answer distribution to compute both the hindsight and best input-only values and test whether trained selectors recover the latter on new prefixes. Across three frozen MoEs, answer information accounts for a provable part of the search gain, while the tested selectors leave input-only value unrecovered; in the controlled reference setting, about 70% of the search gain requires the answer.

1 INTRODUCTION

Mixture-of-experts (MoE) language models use a router to select a small set of experts for each prediction (Shazeer et al., 2017; Fedus et al., 2022). The router makes this choice from the available input, before the next token is known. Once that token is observed, however, evaluating alternative expert choices often reveals one that predicts it substantially better. This contrast suggests that the native router leaves performance unused. It also raises a more basic question: *how much of the improvement found after seeing the answer could a router have obtained before seeing it?*

Recent work has sought better expert use through balanced assignments, more stable routing, and inference-time reselection (Lewis et al., 2021; Dai et al., 2022; Li et al., 2025). Counterfactual evaluations offer a more direct training signal: score alternative expert choices against the realized token, then train the router toward choices that scored better (Yoon et al., 2026; Phenx, 2026). The resulting search gain is tempting to treat as routing headroom. Yet the search and the router solve different decision problems. The search chooses for a particular answer; the router must make one choice that works across the answers that might follow the same input.

This difference creates two explanations for a trained router's shortfall, and they imply different conclusions. Some of the search gain may require knowing the answer: an expert choice that helps one continuation can hurt another, so no decision based only on the input can reproduce the best choice for every continuation. Other gains may remain available before the answer, even if the tested learner fails to identify them. A large search-to-router gap cannot distinguish these cases. Treating the entire gap as a training target asks the router to recover information it does not have; treating a failed learner as evidence that no useful choice exists dismisses value that another learner might recover.

Separating these explanations is difficult when the answer distribution is unknown. A dataset supplies only a few answers to each input, and selecting the best expert choice on those answers already uses information unavailable at decision time. Comparing that choice with a trained router does not

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.

identify the best choice the router could have made. We first address the part that can be established without finding that optimum. For a fixed prefix, we score the same candidate expert choices on independent human completions and compare the best choice for each answer with the best single choice shared by both. Even the shared choice may inspect the two answers. Its comparison with separate choices therefore yields a lower bound on the value of knowing the answer, without assuming that any trained router is optimal.

A lower bound alone cannot tell us how much useful value remains. To measure both sides of the gap, we introduce a second, controlled answer population: the complete token distribution of a separate frozen reference model. At each prefix, this distribution lets us calculate the expected gain of every candidate choice. We can then compute the best choice made before the answer and the average value of the best choice made after it, rather than approximating either with a learner. The same expected gains provide supervision for selectors evaluated on new prefixes. This experiment measures recovery against a known target, although its answer distribution is not an estimate of human responses.

We also ask where recovery breaks down on human data and what a recovered gain represents. Answers from different people let us test whether a useful choice at one prefix survives a change of answer. Holding out entire sentences and texts then tests the harder step: choosing at a new prefix without any answers to that prefix. We compare learned selectors with fixed removal/addition rules fitted on the same data and shuffle their actions across inputs to test whether gains require matching a decision to a particular input. Finally, expert-identity, mixture-weight, case-pooling, and calibration controls examine what the measured score improvement rewards. Figure 1 connects these measurements to the two parts of the gap.

The results show that both explanations matter. On repeated human completions, the paired-answer information bound is positive for OLMoE, Qwen, and Granite, while choices selected on other people's answers also improve prediction. Under the known reference distribution, hindsight search gains 0.1946 nat per token, but the best choice made before the answer gains 0.0571 nat: about 70% of the search gain requires the answer in this controlled population. The tested selectors fail to recover the remaining value on new prefixes despite training on exact expected gains. On human data, an OLMoE selector retains most of its training-prefix gain on new people's answers but little of it on unseen prefixes. Provo refits further show that a transferred gain can largely reflect a shared rule rather than a learned match between inputs and expert choices.

Our contributions are threefold.

- ▷ First, we give a paired-answer measurement that establishes an answer-information advantage without relying on an optimal learned router.
- ▷ Second, we construct a controlled setting in which both the hindsight value and the best input-only value are computable, making learner failure distinguishable from an unattainable target.
- ▷ Third, we trace recovery across answers and inputs and test whether transferred gains require input-specific choices or changes in expert identities. These findings concern fixed sets of single-expert swaps in frozen MoEs, evaluated by next-token log probability. They show why a better choice found after the answer need not produce a better router, while leaving open whether other learners can recover more of the value available before the answer.

2 PRELIMINARY

A sparse MoE layer selects a few experts from a larger collection and combines their outputs. Given an input state h, its routed output is $\sum_{e \in S} w_e E_e(h)$: S contains the selected experts, $E_e(h)$ is an expert's output and w_e is its mixture weight. A shared expert, when present, contributes separately. An expert choice thus specifies both which experts run and how their outputs are weighted.

Let x be a text prefix, y an answer event, and a an expert choice from a finite set $\mathcal{A}_x$ that includes the original router's choice $a_0(x)$. We call this original choice *native*. Write $p_a(y \mid x)$ for the probability of the answer under choice a, and q_x for the target answer distribution. The log-score gain over native is $g_x(a, y) = \log p_a(y \mid x) - \log p_{a_0}(y \mid x)$, in nats. A positive gain means a higher probability for the answer, not higher task accuracy. Our main score pools next-token forms that differ only in ASCII letter case, summing their probabilities without changing the token boundary. Original-token comparisons use one token instead (Appendix B.2).

3 Related Work

Improving routing decisions. Routing methods improve expert use through sparse gating and balanced assignment (Shazeer et al., 2017; Fedus et al., 2022; Lewis et al., 2021). StableMoE separates learning assignments from using fixed assignments (Dai et al., 2022), while later methods revise selection at inference (Li et al., 2025; Su et al., 2026; Hu et al., 2026). Yoon et al. score equal-compute alternatives against a realized token and use that feedback for router-only updates that improve reasoning (Yoon et al., 2026). Our question concerns the information behind that feedback: with candidate predictions fixed, how much does knowing the evaluated answer help?

Information value and repeated outcomes. Choosing after observing an outcome is a classical value-of-information problem (Avriel & Williams, 1970). Chen separates reproducible model advantage from selection over single stochastic generations (Chen, 2026a); related work distinguishes recoverable gain from identifying when to resample or reroute (Chen, 2026b). Our randomness comes from different people's completions of one input (de Varda et al., 2024; Luke & Christianson, 2018), not stochastic model execution; the MoE state and candidate predictions remain unchanged.

Expert identities and expert weights. Mixture-of-Modules changes the composition and order of modules (Gong et al., 2024); Beyond Routing learns how to combine selected expert outputs with their identities fixed (Li, 2026); an exploratory study reports an unsuccessful output-aware selector and hindsight gains from unrelated outputs (Phenx, 2026). Our controls separate identities from weights to test what a learned score improvement rewards.

4 Methodology

4.1 Shared Setup: One Gap with Two Parts

We hold the model and candidate expert choices fixed and vary the information used to choose among them (Figure 1). Human responses provide samples from an unknown answer distribution; a frozen reference model provides a known distribution. In both settings, a selector must act without the evaluated answer. Reference probabilities are used for scoring and supervision, not given to the selector at test time. Every later comparison measures one part of the gap defined below.

Our main candidate set changes routing only at the last MoE layer of the final prefix token: each alternative replaces one of the k native experts with one of the four highest-ranked unselected experts, giving $1+4k$ choices with native. Gate probabilities are rescaled over each selected team to preserve native's total selected weight, so a swap changes expert identities and individual weights but not the active count or total weight. All other layers and positions retain native routing; the learner chooses among these recorded alternatives and does not change their predictions.

To distinguish information from learning, consider two ideal decisions at the same input. An oracle chooses after seeing the answer; an input-only decision must choose before it. Under the target answer distribution q_x, their best expected gains are respectively

$$H(x) = \mathbb{E}_{Y\sim q_x} \max_{a\in\mathcal{A}_x} g_x(a, Y), \qquad C(x) = \max_{a\in\mathcal{A}_x} \mathbb{E}_{Y\sim q_x} g_x(a, Y). \tag{1}$$

Here $C(x)$ is the best value over the candidate set, not the performance of a particular trained network. If a fitted input-only selector achieves $L(x)$, its gap to the oracle has two parts: the information advantage $I(x) = H(x) - C(x)$ and the useful value $C(x) - L(x)$ that it misses. Native is an available choice, so $C(x)$ is nonnegative even when a fitted selector has negative gain, and neither part can be inferred from $L(x)$ alone. The human-response design bounds the first part; the known-reference design measures both parts within its controlled population; the recovery tests of Section 4.4 ask, on human data, where the second part is lost and whether a recovered gain is an input-specific decision at all.

4.2 Bounding Information Value Without an Optimal Learner

Repeated completions give several human answers at one unchanged model state. We score every candidate on every answer and compare separate best choices with the best single choice shared

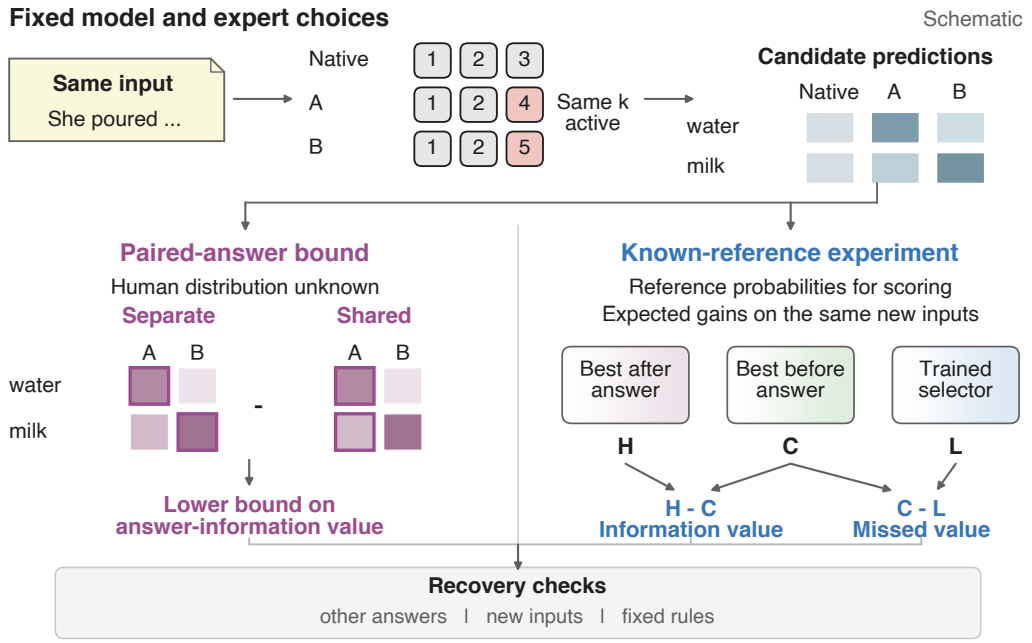


Figure 1: Two measurements of one gap. Top: the model and candidate expert choices stay fixed; only the information used to choose varies. Left: separate versus shared choices on independent human answers lower-bound the information value $H - C$ in expectation. Right: a known reference distribution makes H, C and a frozen learner's value L computable at the same new inputs. Bottom: recovery checks ask where available value is lost on human data and what an obtained gain rewards. The two panels concern different answer populations. Outlines mark selected choices; IDs and cell shades are schematic.

across answers; the shared choice uses the observed answers, so it is not a decision made before they are known.

For two independent answers, compare their separate best choices with the best choice that they must share:

$$W_x(y_1, y_2) = \frac{\max_a g_x(a, y_1) + \max_a g_x(a, y_2)}{2} - \max_a \frac{g_x(a, y_1) + g_x(a, y_2)}{2}. \tag{2}$$

Allowing the shared choice to use both answers makes its expected gain at least $C(x)$. Subtracting this optimistic comparator from the oracle gives $0 \le \mathbb{E}W_x \le I(x)$: a positive expected pair contrast establishes an answer-information advantage regardless of which learner we test. It is a lower bound, not the fraction of the total gap caused by information.

The confidence calculation must account for each person answering several prefixes. We form disjoint pairs of people within each data list, without using their answers or missing-response patterns; each pair contributes its mean W_x over that list's fixed prefixes, capped at 0.1 nat, and a bounded-mean inequality gives the 95% lower bound. It assumes independent, exchangeable respondent records within each list, allows dependence across one person's answers and across lists, and covers the fixed prefixes and candidate choices (proof and missing-answer convention: Appendix A.1).

4.3 Measuring Both Values with a Known Reference

The paired-answer design leaves the best input-only value unknown. To measure it directly, we replace the unknown human distribution with the predictions of a separate frozen reference model. At each prefix, we score every candidate over the complete aligned vocabulary. Averaging each candidate's gain under the reference probabilities and then taking the maximum gives $C(x)$; taking the maximum separately for each token before averaging gives $H(x)$. Both values and their difference are thus computable without fitting a router. This is a controlled answer population, not an estimate of the human population; its values cannot be added to the human-response bounds.

For learning, the target at each training prefix is the vector of exact expected gains over the candidate choices. This removes noise from sampling a few answers while retaining the need to generalize to new inputs. We fit input-state, expert-output and readout-based selectors on separate documents, choose checkpoints by validation gain, and evaluate their frozen choices on new prefixes. Comparison with $C(x)$ measures the value these procedures miss; comparison with native tests whether they improve prediction at all (details: Appendix D).

4.4 Testing Recovery and What a Recovered Gain Requires

Two questions remain about the second part of the gap: whether the tested learners recover available value at new prefixes, and whether the gains they do obtain there need input-specific decisions at all. On human data, answers from a separate *support group* establish useful choices without knowing the input-only optimum. At each prefix, we select by mean support score and test that choice on other people. Its expected evaluation gain is at most $C(x)$. A learner must then find useful choices at new prefixes without their support answers.

Our main learner is a ridge regressor trained on other texts. It takes the pre-MoE state and centered gate logits, predicts candidate mean scores and chooses the highest. The MoE stays fixed. Splits keep whole texts separate, and validation score selects ridge strength; no answers to test prefixes enter selection.

We evaluate the same fitted selector on new people at its training prefixes (transfer across answers) and on unseen prefixes (transfer across inputs as well), and separately vary training-text count and support answers per prefix with the learner, folds and validation feedback fixed.

For removal, we index the native team by expert ID (*ID-sorted slots*) or gate score (*gate ranks*); additions remain gate-ranked. This changes action coordinates, not candidates. A fixed rule fits one removal/addition position pair on the same training data. A second control shuffles the learner's rank actions within each test fold, preserving their frequencies. The native gate still maps ranks to input-specific experts: shuffling removes learned rank matching, not all input dependence.

We also fit the pretrained gate with EPO-style gain-weighted preferences (Yoon et al., 2026), changing selection within the fixed candidate set, not execution weights (Appendix C.8), and test alternative training targets and expert-output features. Score controls hold choices fixed while changing case pooling or temperature, to identify gains from capitalization or confidence.

Interpreting recovered gains. A swap changes both expert identities and mixture weights. We separate these at the same removal/addition index: identity-only gives the incoming expert the removed expert's original weight; weight-only keeps native experts but assigns the joint swap's incoming weight to the original expert at that slot; joint changes both and native neither.

5 Experiment

5.1 Evaluation Conditions

We use OLMoE-1B-7B-0924, Qwen3-30B-A3B-Base and Granite-4.0-H-Tiny (Muennighoff et al., 2025; Qwen Team, 2025; IBM Granite Team, 2025). The first two activate eight routed experts and have 33 local choices; Granite uses six plus a shared FFN and has 25 choices. We score the first model token of each human response word using the case-pooled event in Section 2. Unless stated otherwise, temperature is $T = 1$ and gains are in nats relative to native routing on the same score.

UCL supplies repeated completions at the same inputs: 205 English sentences (de Varda et al., 2024). After execution checks and a 40-sentence pilot, evaluation uses 917 prefixes for OLMoE and Qwen, and 903 for Granite. Each of eight lists has 24 support respondents and 52–56 evaluation respondents; the latter form disjoint pairs for information measurement. The pilot supplies 230 fitting and validation prefixes (228 for Granite); splits keep sentences separate. Scores average answers within each prefix, then weight prefixes equally (Appendix F).

Provo tests transfer beyond UCL, with 63,258 response counts at 1,581 prefixes from 46 texts (Luke & Christianson, 2018). We first evaluate UCL-trained policies without Provo feedback, then refit with whole texts held out. Its aggregate counts lack respondent identities, precluding the person-

pair bound, and later controls reuse previously inspected panels as secondary analyses outside the bound's sampling guarantee (Appendix F).

A separate controlled study keeps OLMoE-0924 as the routed model and uses OLMoE-1B-7B-0125-Instruct's fixed answer distribution on FineWeb-Edu prefixes (Penedo et al., 2024): a known target distribution, not a better model of human answers. We train with 994 or 9,893 documents and evaluate 1,968 new prefixes, scoring original tokens, with all policy choices frozen before test access.

5.2 Does the Search Gain Require the Answer?

Answer-aware search can gain even when no better input-only decision exists. Under native's own predicted answer distribution native routing already maximizes expected log score, yet search gains 0.1832 nat in OLMoE and 0.1739 in Qwen on full-vocabulary original-token scores (Appendix A.2), entirely from answer information. Human responses need not follow that distribution, so Table 1 measures the advantage and the gains that transfer across human answers.

Table 1: Answer access and cross-answer gains on UCL. OLMoE and Qwen use 917 prefixes and 33 choices; Granite uses 903 and 25. Learned selector is the original ID-sorted slot ridge. The final row is a lower bound from independent answer pairs, not a confidence interval for the observed gap above it. Shared and support choices use answers at each evaluation input.

Choice	Available when choosing	OLMoE	Qwen	Granite
Observed gains over native				
Best per answer	Evaluated answer	0.2213	0.2131	0.1326
Best mean on same answers	Evaluation group	0.0734	0.0848	0.0453
Chosen on other answers	Other people, same prefix	0.0562	0.0689	0.0353
Learned selector	Prefix only; trained elsewhere	0.0046	0.0148	-0.0053
Observed gap: best per answer minus best mean		0.1480	0.1283	0.0873
Paired-answer information bound (95%)		**0.0248**	**0.0180**	**0.0017**

UCL shows both an answer-information advantage and useful cross-answer choices. OLMoE's oracle gains 0.2213 nat, the best mean choice on the same answers 0.0734, and selection on other answers 0.0562; these vary answer access, not stages of router training. The pair analysis attributes at least 0.0248 nat to answer information at 95% confidence rather than to learner failure; the observed oracle-minus-shared gap is larger, but finite-sample fitting prevents reading it as the exact population value. Analogous bounds are positive in Qwen and Granite, 0.0180 and 0.0017 nat: part of a better choice's value is the answer itself.

5.3 How Much Value Remains Without the Answer, and Is It Recovered?

The second design makes both values known under a separate reference distribution (Section 4.3). Answer-aware search gains 0.1946 nat on new prefixes, while the best input-only choice gains 0.0571. Their difference, 0.1375 nat (95% interval $[0.1321, 0.1432]$), is the information advantage in this controlled population. About 70% of the search gain thus requires the answer; the remaining 0.0571 nat is a known target that an input-only selector could attain (Figure 2).

Exact expected-gain supervision does not guarantee recovery. The state-based MLP gains 0.0463 nat on its 994 training documents but loses 0.0119 on new prefixes; with 9,893 training documents, it loses 0.0050. Adding expert outputs, the residual state, or scoring through the model's output layer, as either a generic or a structured readout, also leaves mean test gains below zero. Checkpoints are selected by validation gain at 200 or 800 updates; in the larger State MLP runs, further fitting raises training gain but lowers validation gain (Appendix D).

The known optimum makes the negative result informative: useful input-only choices exist in this controlled population, but these procedures miss them. Answer information cannot explain the failure, and since supervision is exact, neither can noise from sampled answers: the selectors fit their training prefixes and do not carry the choices to new ones.

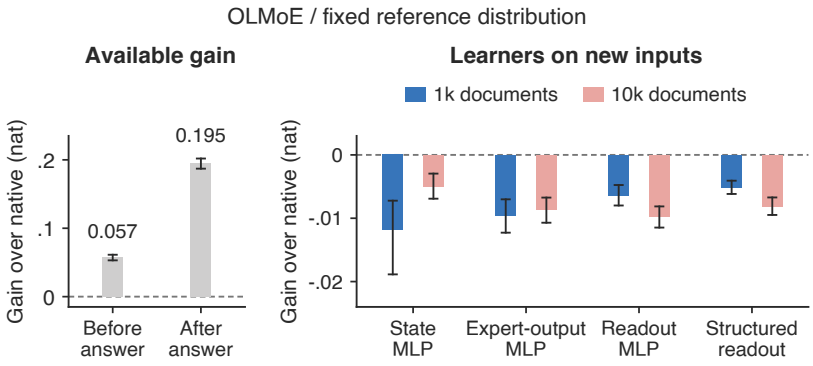


Figure 2: Known opportunity and frozen test performance under a fixed reference distribution. Left: best choices before and after seeing the answer. Right: mean gains of three separately trained seeds on the same 1,968 new prefixes (different scales); 95% prefix-bootstrap intervals are conditional on the fitted policies. Budgets are 994 and 9,893 documents; all families: Appendix D.

5.4 Human Data: Value Survives New Answers, Not New Inputs

More support answers improve selection for other people. With one answer, OLMoE gains 0.2196 nat on that answer but only 0.0083 on other people; with 24, the latter gain reaches 0.0562. The other models show the same trend (Figure 3). A new-text learner must identify useful choices without support answers at its test prefixes.

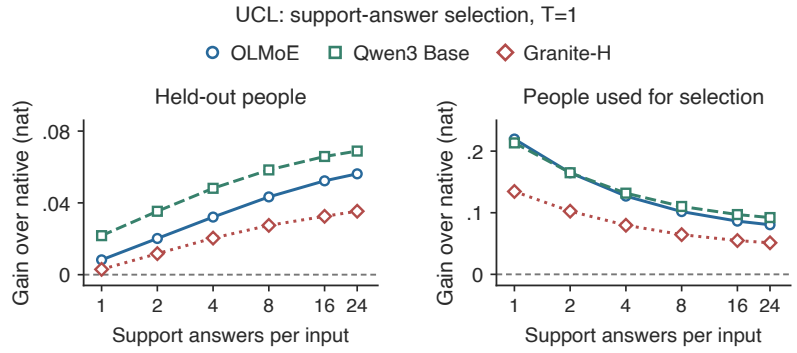


Figure 3: More support answers improve gains on other people's answers, while the score on the people used for selection falls as one choice must serve more answers. Candidates and evaluation people stay fixed; panels use different vertical scales. UCL, gains over native at $T = 1$.

Useful choices thus survive a change of people; can a trained selector also carry them to new inputs? We hold one OLMoE ridge selector (ID-sorted slots) fixed after training on 230 UCL prefixes. Its gain is 0.0808 nat on training answers, 0.0606 on other people's answers to those same prefixes, and 0.0046 on unseen UCL prefixes (Figure 4). The last gain beats both fixed rules but is far below selection using support answers at the new prefixes: useful choices remain, while learned recovery falls (Appendix C.1).

More answer feedback helps this UCL learner, but its new-input gains remain small. With 165–166 training sentences, raising support answers from six to 24 lifts mean gain from 0.0020 to 0.0064 nat (validation held at 24 answers); the 40-sentence budget shows the same direction, more texts have mixed effects across folds, and all four means exceed matched fixed rules (Appendix C.3). Neither alternative output coordinates nor feature compression beats full-feature ID-sorted slots at either budget (Appendix C.3).

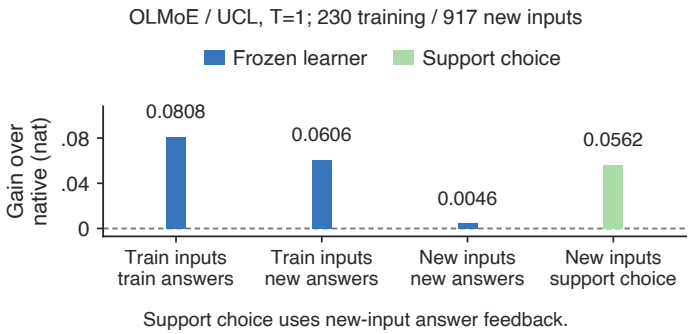


Figure 4: The frozen selector's gain survives new answers better than new inputs. The first three bars evaluate the same OLMoE selector trained on 230 prefixes. The last bar instead uses support answers at each of the 917 unseen prefixes to select a choice, then tests it on other people. Fixed-rule comparison: Appendix C.1.

5.5 Provo: What Transfers to New Texts Is Mostly a Shared Rule

If learners lose value at new inputs, what do they carry there? Changing the text source to Provo without retraining, the UCL-trained slot selectors gain 0.0011 nat in OLMoE and 0.0032 in Qwen, below UCL-selected fixed rank rules' 0.0377 and 0.0225. To separate this domain shift from learning within Provo, we refit selectors and rules on Provo training texts, holding out whole test texts, and ask whether the refits' gains depend on matching rank actions to inputs (Figure 5).

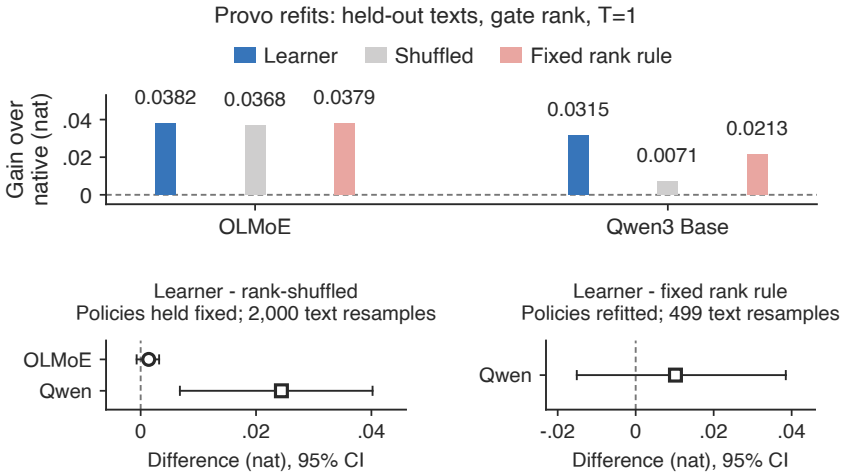


Figure 5: Provo refits: gain over native, input matching and fixed-rule comparison. Case-pooled scores at $T = 1$; rules use the same training data. The 95% whole-text intervals differ: selector-minus-shuffle fixes fitted policies, while Qwen selector-minus-rule refits the selector and reselects the rule, and spans zero (Appendices C.4, C.5).

Within these Provo refits, changing the removal coordinate from ID-sorted slots to gate ranks raises OLMoE's gain from 0.0101 to 0.0382 nat with candidates and training features fixed, yet most of the gain survives shuffling and is matched by the fixed rank rule. Even the slot selector's larger shuffle gap (0.0137) needs a control: a fixed slot rule can beat its rank shuffle because one ID-sorted slot maps to different gate ranks across inputs (Appendix C.4).

Qwen shows input matching, but its advantage over the fixed rule remains uncertain. Its gate-rank selector beats shuffled rank actions by 0.0244 nat; with fitted policies fixed, 95% whole-text bootstrap intervals over shuffle are $[-0.0007, 0.0032]$ for OLMoE and $[0.0068, 0.0402]$ for Qwen. Qwen exceeds its matched rule by 0.0102 nat, but resampling whole texts while refitting the selector and reselecting the rule gives an approximate 95% interval of $[-0.0151, 0.0385]$ (Appendix C.5): input matching does not establish reliable superiority once training variability is included.

Training on calibrated scores reverses Qwen's ordering against the rule: at the UCL-selected evaluation temperature $T = 1.25$, retraining the selector on $T = 1.25$ instead of $T = 1$ targets lowers its gain from 0.0182 to 0.0093 nat, while the matched rule gains 0.0133 in both cases and native is subtracted at the same evaluation temperature (Appendix C.5). Granite's tested selectors lose to native on UCL and frozen Provo transfer despite positive cross-answer opportunity (Appendix C.4). Input matching and beating a matched rule remain separate tests, and the two models fail different ones: OLMoE's transferred gain is mostly a preference shared across inputs, while Qwen does match inputs but does not reliably improve on a rule that needs no such matching.

5.6 Do Objectives, the Native Gate or Expert Outputs Help?

Three remedies for the lost value are tested next. First, the target: on UCL, selecting with the same 24 support answers by mean log score rather than by the most frequent per-answer oracle winner gains 0.0562 nat on other people's answers in OLMoE versus 0.0177, and Qwen and Granite also favor mean score (Appendix C.6); winner labels discard value before learning begins.

Second, the learner: we replace the added regressor with a trainable copy of each model's original gate, using gain-sensitive preferences over the same candidates with execution weights fixed. These gates recover best-mean choices on all 230 UCL fitting prefixes, yet Qwen gains 0.0114 nat on held-out UCL prefixes, above its rank rule, but loses 0.0047 on Provo without refitting; OLMoE is below native on both, and Provo refits leave both below matched rank rules (Appendix C.8). A nonlinear direct-utility scorer helps only small-data OLMoE, mainly in one fold (Appendix C.7): better fitting of preferred choices does not ensure transfer.

Third, the features: candidate expert outputs do not improve the Provo refit comparison. Matched scorers augment OLMoE's ridge with complete, compressed, scrambled or no output vectors, holding folds, pre-execution features and training schedules fixed. Executing twelve experts before selecting eight, the full-output scorer gains 0.0275 nat on new texts, below the no-output control's 0.0329 and the ridge's 0.0382. After all-data refitting it beats its own ridge by 0.0248 nat on training prefixes but loses 0.0107 on held-out texts, and the before-refit numbers order the same way: output features improve fitting, not transfer (Appendix C.9).

5.7 What a Recovered Gain Rewards

A recovered gain need not require different expert identities. At the swap indices selected by the fixed UCL rank rule, replacing one expert on OLMoE's Provo candidates, identities alone gain 0.0350 nat, weights alone 0.0373 and the joint swap 0.0377: either operation recovers most of the joint gain, and their effects are not additive (Appendix E.1). Selection also affects attribution. A second candidate set exchanges whole native teams, expert IDs and weights together, between UCL prefixes, which preserves every physical expert call (Appendix E.2). There, selecting for weights alone rather than reusing joint-selected assignments raises held-out gain from 0.0152 to 0.0445 nat against 0.0949 for joint, with evaluation people separate (Appendix E.3).

Score controls. Policies selected on original-token scores gain largely by shifting probability between case variants: with actions held fixed, this accounts for 0.0812 of Qwen's 0.0926 nat, and OLMoE shows the same pattern (Appendix B.2); the main case-pooled score removes this effect. Even then, native calibration to the UCL-selected $T = 1.25$ gains 0.3284 nat in Qwen and 0.2647 in Granite on Provo without changing experts (Table 4), so interpreting learned recovery requires the score and calibration to be held fixed, as in our same-temperature comparisons.

6 Conclusion

Within a fixed candidate set on frozen MoEs, we find two reasons better expert choices need not make better routers. Part of a choice found after the answer needs that answer: paired human completions establish this advantage without assuming an optimal learner, and under a known reference it is about 70% of the search gain. The rest is available but unrecovered: the tested learners keep it across new answers, lose it at new prefixes, and miss it even with exact expected-gain supervision, never beating a fixed rule that needs no input-specific decision. Search thus overstates a router's target while a failed learner understates what remains.

AI USE STATEMENT

This work involved limited use of large language models solely for language refinement purposes. The models were used to improve grammar and typos. The content was written entirely by the authors. No parts of the manuscript, including conceptual content, technical descriptions, data analysis, or results, were generated by any GenAI or LLMs. All scientific ideas, analyses, and conclusions are the sole responsibility of the authors.

ETHICS STATEMENT

We reuse the UCL and Provo sentence-completion data, FineWeb-Edu text, and released model checkpoints. No new participants were recruited and no new human responses were collected. Respondent identifiers serve statistical grouping, pairing and resampling; the analysis concerns aggregate model performance, not individual participants. The data and pretrained models can carry linguistic and social biases. Results on short English prefixes do not establish performance across languages or demographic groups. A higher response score also does not establish safer or fairer behavior.

REPRODUCIBILITY STATEMENT

The Methodology section defines the candidate expert choices, scoring and information comparisons. Appendix A gives the assumptions and proofs, and Appendix B records data processing and execution settings. Appendices C and D specify training, validation, held-out evaluation and the controlled reference experiment. Appendices E and F document the controls, numerical checks, unsuccessful branches and scope. The accompanying supplement includes an indexed archive of detailed results and a record of all 60 frozen policies from the reference experiment, with their seeds, training budgets, selected checkpoints and evaluation scores. Code and analysis scripts are available at https://anonymous.4open.science/r/MoE-Routers-Choices-Repro.

REFERENCES

M. Avriel and A. C. Williams. The Value of Information and Stochastic Programming. *Operations Research*, 18(5):947–954, 1970. doi: 10.1287/opre.18.5.947. URL https://pubsonline.informs.org/doi/10.1287/opre.18.5.947.

Yoshua Bengio and Yves Grandvalet. No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004. URL https://jmlr.org/papers/v5/grandvalet04a.html.

Teng-Ruei Chen. How Much of the Routing Gap Is Real? Decomposing the Router-to-Oracle Gap into Reproducible Specialist Advantage and Single-Draw Label Noise. arXiv preprint arXiv:2607.03436, version 2, 2026a. URL https://arxiv.org/abs/2607.03436v2.

Teng-Ruei Chen. Resample or Reroute? Recoverable Stopping Debt Without Identified Action Selection. arXiv preprint arXiv:2607.08665, version 3, 2026b. URL https://arxiv.org/abs/2607.08665v3.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable Routing Strategy for Mixture of Experts. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7085–7095, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.489. URL https://aclanthology.org/2022.acl-long.489/.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56:5190–5213, 2024. doi: 10.3758/s13428-023-02261-8. URL https://link.springer.com/article/10.3758/s13428-023-02261-8.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Zhuocheng Gong, Ang Lv, Jian Guan, Wei Wu, Huishuai Zhang, Minlie Huang, Dongyan Zhao, and Rui Yan. Mixture-of-Modules: Reinventing Transformers as Dynamic Assemblies of Modules. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20924–20938, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1164. URL https://aclanthology.org/2024.emnlp-main.1164/.

Wentao Hu, Yanbo Zhai, Xiaohui Hu, Mingkuan Zhao, Shanhong yu, Xue Liu, Kaidong Yu, Shuangyong Song, and Xuelong Li. Awakening Dormant Experts:Counterfactual Routing to Mitigate MoE Hallucinations. 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. 47257–47271, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2187. URL https://aclanthology.org/2026.acl-long.2187/.

IBM Granite Team. Granite-4.0-H-Tiny. Hugging Face model repository, 2025. URL https://huggingface.co/ibm-granite/granite-4.0-h-tiny. Model card and instruction-tuned checkpoint; distinct from Granite-4.0-H-Tiny-Base.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE Layers: Simplifying Training of Large, Sparse Models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274, 2021. URL https://proceedings.mlr.press/v139/lewis21a.html.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. R2-T2: Re-Routing in Test-Time for Multimodal Mixture-of-Experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35292–35316, 2025. URL https://proceedings.mlr.press/v267/li25bc.html.

Zongfei Li. Beyond Routing: Decoupling Expert Dispatch and Aggregation in Sparse Mixture-of-Experts. Preprint, arXiv:2608.08853, 2026. URL https://arxiv.org/abs/2608.08853.

Steven G. Luke and Kiel Christianson. The Provo Corpus: A large eye-tracking corpus with predictability norms. *Behavior Research Methods*, 50:826–833, 2018. doi: 10.3758/s13428-017-0908-4. URL https://link.springer.com/article/10.3758/s13428-017-0908-4.

P. Massart. The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality. *The Annals of Probability*, 18(3):1269–1283, 1990. doi: 10.1214/aop/1176990746. URL https://doi.org/10.1214/aop/1176990746.

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 Smith, Pang Wei Koh, Amanpreet Singh, and Hanna Hajishirzi. OLMoE: Open Mixture-of-Experts Language Models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 62061–62121, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/9b224ace8963c9385ad5e2b5c9039b97-Paper-Conference.pdf.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Phenx. Why Couldn't an LLM Pick Its Best Expert? Exploratory research blog, August 12, 2026. URL https://www.phenx.ai/research/why-couldnt-an-llm-pick-its-best-expert.

Qwen Team. Qwen3-30B-A3B-Base. Hugging Face model repository, 2025. URL https://huggingface.co/Qwen/Qwen3-30B-A3B-Base. Model checkpoint and model card.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Guinan Su, Yanwu Yang, Li Shen, Lu Yin, Shiwei Liu, and Jonas Geiping. Rewiring Experts on the Fly: Continuous Rerouting for Better Online Adaptation in Mixture-of-Expert Models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/60869.

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.

GUIDE TO THE APPENDIX

Appendix A gives the information-bound proof; B records data and scoring; C gives transfer and learner protocols; D details the known-reference experiment; E separates expert identities, weights and score effects; and F records selection history and scope. Complete historical grids, per-fold results and execution ledgers are retained in the accompanying full-results PDFs. The supplement's index locates them by topic.

A MEASUREMENT AND STATISTICAL DETAILS

A.1 THE PAIRED-ANSWER BOUND

For independent $Y_1, Y_2 \sim q_x$, the first term of Equation 2 has expectation $H(x)$. Convexity gives

$$\mathbb{E} \max_a \frac{g_x(a, Y_1) + g_x(a, Y_2)}{2} \geq \max_a \mathbb{E} g_x(a, Y) = C(x),$$

so $0 \leq \mathbb{E} W_x(Y_1, Y_2) \leq I(x)$. This is the step that removes the need to approximate the best input-only policy with a trained learner.

For list ℓ, let X_ℓ be its fixed prefixes and $P_{\ell j}$ its jth disjoint respondent pair. We shuffle respondent IDs with a fixed seed and pair adjacent evaluation IDs, independently of answers and missingness. Define $Z_{\ell j} = \min\{b, |X_\ell|^{-1} \sum_{x \in X_\ell} W_x(P_{\ell j})\}$. Capping after averaging across prefixes gives $Z_{\ell j} \in [0, b]$ and can only decrease its expectation. With m_ℓ pairs, cap $b = 0.1$ nat and error allocation $\alpha_\ell = 0.05/8$, Hoeffding's inequality gives

$$L_\ell = \max \left\{ 0, \overline{Z}_\ell - b \sqrt{\frac{\log(1/\alpha_\ell)}{2m_\ell}} \right\}. \tag{3}$$

Independent, exchangeable respondent records within a list make this a $1 - \alpha_\ell$ lower bound on that list's mean information value. A union bound across lists, followed by weighting with $|X_\ell| / \sum_j |X_j|$, gives the panel bound. Dependence within one person's responses and reuse of people across lists are allowed; prefix-response pairs are not treated as independent samples.

The confirmation panel has eight missing support responses and one missing evaluation response. For the witness, a nonresponse has zero utility for every action and stays in the denominator; descriptive gains instead average valid responses within each prefix. This fixed augmented-response convention adds no preferred action. A zero bound is not evidence that information value is zero. Even a zero population pair loss can coexist with positive information value when actions tie; with an almost-surely unique best action, zero pair loss implies a common best action and hence $I(x) = 0$.

A.2 NATIVE-DISTRIBUTION CONTROL

When answers follow native's distribution, $q_x = p_{a_0}$,

$$\mathbb{E}_{p_{a_0}} g_x(a, Y) = -D_{\mathrm{KL}}(p_{a_0} \| p_a) \leq 0. \tag{4}$$

Native is an available action, so $C(x) = 0$. Search still gains 0.1832 nat in OLMoE and 0.1739 in Qwen, computed over the full vocabulary. These gains are entirely answer-information value in this control distribution, not estimates of that fraction in human data. More generally, a best common action a^* gives $I(x) = \mathbb{E}[\max_a g_x(a, Y) - g_x(a^*, Y)]$; the integrand is nonnegative, and information value is zero exactly when some common action is best almost surely.

A.3 SENSITIVITY OF THE CONFIDENCE BOUND

At the fixed OLMoE middle layer, original case-pooled pair means of 0.0248–0.0263 nat lie below Hoeffding penalties of 0.0321–0.0333. We retain this zero-bound result and make a secondary empirical-CDF comparison. For the empirical CDF $\widehat{F}_\ell$ of the same capped pair averages, the DKW–Massart band (Massart, 1990) implies

$$\mathbb{E} Z_\ell \geq L_\ell^{\mathrm{CDF}} = \int_0^b [1 - \widehat{F}_\ell(t) - \varepsilon_\ell]_+ \, dt, \qquad \varepsilon_\ell = \sqrt{\frac{\log(2/\alpha_\ell)}{2m_\ell}}. \tag{5}$$

Integrating the whole band avoids choosing a favorable threshold and allows atoms and ties. The joint comparison allocates $0.05/128$ to each list, condition and method: eight lists, eight model/layer/score conditions and two methods. It retains all conditions and the original cap, with no independence required across conditions or methods. The within-list respondent assumptions remain those above. This secondary fixed-panel analysis does not replace the original bound's coverage statement.

Table 2: Information lower bounds in nats. Original allocations and the joint 95% family have different coverage; both methods and every condition are retained.

Intervention	Response	Original Hoeffding	Joint Hoeffding	Joint CDF
OLMoE final	Raw	0.0300	0.0226	0.0314
OLMoE final	Class	0.0248	0.0173	0.0306
Qwen final	Raw	0.0231	0.0156	0.0267
Qwen final	Class	0.0180	0.0106	0.0268
Granite final	Raw	0.0036	0.0000	0.0183
Granite final	Class	0.0017	0.0000	0.0172
OLMoE middle	Raw	0.0000	0.0000	0.0144
OLMoE middle	Class	0.0000	0.0000	0.0141

Table 2 shows which zeros reflect the precision of the original certificate. The survival integral and an independent quantile integral agree within 2.78×10^{-17} nat over all 128 comparisons. The original temperature-grid sensitivity divides each list's error budget by seven; its complete results and descriptive shortfall accounting are in the full-results supplement.

B Data and Shared Protocols

B.1 Execution and Held-Out Processing

OLMoE runs in FP32; Qwen uses BF16 prefix blocks and an FP32 final MoE block/readout. Their predictors concatenate pre-MoE states and centered gate logits (2,112/2,176 coordinates). Sentence-group validation selects ridge penalties from $\{0.1, 1, 10, 100\}$ by mean response score for either training target; predictions precede test scoring. The native-temperature grid is $\{0.5, 2/3, 0.8, 1, 1.25, 1.5, 2\}$. Score normalization and aggregation use GPU FP64.

The five sentence-held-out folds retain one pilot validation set and nested budgets, including all 40 pilot sentences and ultimately every eligible non-test sentence. Fitting, standardization and penalty selection exclude the test fold; evaluation answers never train policies or rules. Each prefix contributes one held-out choice per budget. Eight support and one evaluation response are missing: descriptive gains average valid answers, whereas pair witnesses give missing answers zero utility for every action and retain fixed prefix denominators.

B.2 Case-Pooled Events

Fix the original first-token boundary before mapping decoded token bytes to ASCII lowercase, inverting the ByteLevel alphabet first. Pool all ordinary vocabulary IDs; special, added and unassigned IDs stay singleton. OLMoE/Qwen have 40,934/128,128 classes from 50,304/151,936 output IDs. For class c, $p_a^c(c \mid x) = \sum_{v:c(v)=c} p_a(v \mid x)$ uses the complete-vocabulary normalizer, never observed-answer renormalization. All observed-class members are retained; forced forwards check projected scores within 10^{-5} nat. Lowercasing the word before tokenization can change its boundary and is a different experiment, not full-word scoring.

For identical actions, responses and averaging weights, factorization gives $g = \log(p_a^c/p_{a_0}^c) + \log[p_a(y \mid c, x)/p_{a_0}(y \mid c, x)]$: class gain plus within-class gain. The original raw-trained mean-score policies give $0.0439 = 0.0043 + 0.0396$ nat in OLMoE and $0.0926 = 0.0114 + 0.0812$ in Qwen. Most gain concerns case forms. Separately class-trained gains are 0.0046/0.0148; winner-label gains are $-0.0075/0.0028$. These reselections cannot be substituted into the fixed-action identity, nor can independently maximized oracle values.

B.3 Provo Qualification

Requiring valid ASCII prefixes/response words, consistent denominators and unambiguous aligned rows, 1,658 of 2,687 Provo groups pass source checks and 1,581 pass both tokenizers, retaining 63,258 responses across 46 texts. Rejection reasons include inconsistent totals, duplicate or misaligned rows, non-ASCII or empty prefixes, empty responses and inconsistent metadata; counts are never repaired or redistributed. The other 77 candidates have fewer than four prefix tokens; retained lengths are 4–76/4–74 for OLMoE/Qwen, without duplicate inputs or exact UCL overlap. Score the first token of the first whitespace-delimited response word; appending the response must preserve both tokenizers' prefix sequences. Multiplicities define empirical histograms, not respondent records or population confidence bounds.

B.4 Granite Hybrid Checkpoint

Granite-4.0-H-Tiny is instruction-tuned: 7B parameters, 40 hybrid layers, 64 routed experts, top-six selection and a shared FFN. FP32 raw-cloze execution changes only the final token's final routed mixture, retaining the shared FFN, Mamba state, residual scale and readout. Native plus 24 swaps use four unselected experts. Sixteen inputs shorter than four tokens leave 228 pilot/903 test prefixes, retaining the 40/157 sentence split and original people; all Provo prefixes qualify. Its 100,352 IDs form 76,856 classes. Features are 1,536 states plus 64 gate logits; 165/63 pilot prefixes fit/validate, then all 228 refit. Slot/rank penalties are 10/100 in both scores; no winner-label or expanded learner search is added.

C Transfer and Learning Comparisons

C.1 Cross-Answer and Cross-Input Evaluation

Support selection averages scores over the same first $1, 2, 4, 8, 16, 24$ support people, with membership fixed across a list's prefixes. All choices are then scored on the unchanged evaluation group. For Figure 4, the original 230-prefix OLMoE selector and both fixed rules remain frozen; only the evaluation people or inputs change. Missing pilot answer scores were acquired with all original candidates, weights and score definitions fixed, rather than counted as nonresponses. The original 917 test choices replay exactly. On unseen inputs the ID-sorted slot and rank rules gain -0.0007 and -0.0066 nat, below the selector's 0.0046. The two input panels are different texts, not a randomized intervention on training coverage.

C.2 Provo Transfer, Refitting and Coordinates

Frozen transfer applies the UCL-selected policy and rules without Provo feedback. Within-Provo refits instead shuffle the 46 texts into five fixed outer folds, keeping every test text out of preprocessing, fitting and selection. Ten non-test texts supply the validation pool; input-only sampling selects 64 validation and 166 training prefixes from different texts. The 230-prefix condition selects ridge strength, then refits on all 230. The larger condition fits all prefixes outside test and validation texts, selects on the same 64 validation prefixes, then refits on all non-test texts (1,179–1,334 prefixes). Both models use penalties $0.1, 1, 10, 100$, with larger penalties preferred on ties. The fixed rules use the same refit labels. All 1,581 test prefixes are evaluated once, retaining their response counts.

Coordinate comparisons keep the 33 physical candidates, inputs, ridge loss and parameter count fixed. An input-only permutation replaces ID-sorted removal slots with gate ranks; additions remain gate-ranked. Gate ties use physical IDs and decision ties use the first output, with native first. The permutation changes the learned mapping to physical actions, not their execution. Source transfer and Provo refitting differ in training texts and response feedback, so their contrast alone does not isolate a causal effect of domain shift. These are secondary comparisons on previously inspected panels.

C.3 Answer Feedback and Feature Compression

The OLMoE feedback comparison crosses 40 training sentences (230 prefixes) or 165–166 sentences (949–971 prefixes) with six or 24 support answers. Five sentence-held-out folds cover the

same 917 evaluation prefixes. The six-answer condition partitions each list's support people into four fixed groups and averages the four fitted policies' utilities, not their predictions. Validation always uses 24 answers on the same ten pilot sentences, followed by refitting with the declared training-answer panel. Table 3 retains all four conditions. More texts has mixed fold-level effects; more answers helps both aggregate budgets, but does not restore the support-choice gain.

Table 3: UCL feedback budgets: held-out gains over native in nats. Rules use the same training answers; six-answer rows average four policies.

Training sentences	Answers	Ridge	Slot rule	Rank rule
40	6	0.0022	-0.0032	-0.0040
40	24	0.0046	-0.0007	-0.0066
165–166	6	0.0020	-0.0016	-0.0023
165–166	24	0.0064	-0.0034	-0.0009

The representation comparison crosses these text budgets with slot or rank outputs and full 2,112-dimensional features or PCA to 32 or 128 dimensions. All 12 conditions retain the 24-answer targets, candidates and ridge grid. Centering, scaling and PCA fit only training sentences, then refit on allowed training and validation sentences before test prediction. PCA uses orthonormal directions without whitening. No alternative exceeds full-feature slots at its text budget, although PCA32 improves the larger-budget rank mean from 0.0016 to 0.0024 nat. Fold-level effects vary. The full grid and all folds remain in the full-results record; these summaries do not select a preferred condition using test outcomes.

C.4 Input Matching and Fixed Rules

For each test fold, the shuffle control preserves the fitted policy's frequency of every local gate-rank action and computes its expected score over permutations of those labels across inputs, including fixed points. No policy is retrained. The recipient's native gate still maps ranks to physical experts, so this removes the learner's rank matching, not all input dependence or physical expert-call differences. Even a fixed ID-sorted slot rule can beat its rank shuffle: at the 40-sentence OLMoE budget it gains -0.0007243 nat versus -0.0042784 when shuffled. A slot can map to different gate ranks across inputs. This is why matching and improvement over a training-matched fixed rule are separate comparisons.

Table 4 retains all three models and the frozen/refitted distinction. On all-data Provo, 2,000 within-fold whole-text bootstrap draws give selector-minus-shuffle intervals of $[0.0072, 0.0205]$ for OLMoE slots, $[-0.0007, 0.0032]$ for OLMoE ranks and $[0.0068, 0.0402]$ for Qwen ranks. These intervals condition on fitted policies and observed answers. Qwen's conditional selector-minus-rule interval, $[0.0026, 0.0175]$, does not include training variability; the refitted comparison below does.

Table 4: Coordinate and calibration controls. Case-pooled gains in nats over native at $T = 1$. Frozen Provo uses UCL training; Provo refits hold out whole texts. The rank rule uses the selector's training data. Native T^* changes only confidence, using the UCL-selected temperature.

		Selector outputs			
Model	Test condition	ID-sorted slot	Gate rank	Rank rule	Native T^*
OLMoE	UCL	0.0046	-0.0007	-0.0066	0.0000
OLMoE	Provo, frozen	0.0011	0.0101	0.0377	0.0000
OLMoE	Provo, refit	0.0101	0.0382	0.0379	–
Qwen Base	UCL	0.0148	0.0099	0.0011	0.0320
Qwen Base	Provo, frozen	0.0032	0.0218	0.0225	0.3284
Qwen Base	Provo, refit	0.0066	0.0315	0.0213	–
Granite-H	UCL	-0.0053	-0.0019	-0.0045	0.0495
Granite-H	Provo, frozen	-0.0123	-0.0012	-0.0024	0.2647

C.5 Calibration and Full-Refit Uncertainty

At fixed Provo evaluation temperature $T = 1.25$, Qwen's rank selector gains 0.0182 nat when trained at $T = 1$ and 0.0093 when trained at $T = 1.25$; the corresponding fixed rule gains 0.0133. Native is subtracted at the same evaluation temperature in each comparison. Both training budgets and all temperature pairs are retained in the full-results record. This observed reversal motivates uncertainty over the fitting procedure, rather than only over scores of an already fitted policy. Overlapping fold training sets are not independent learning replications (Bengio & Grandvalet, 2004).

Each of 499 bootstrap draws (seed 2026091801) resamples whole texts within their original outer test folds. A text's multiplicity is shared wherever it occurs in training, validation or evaluation; copies never cross the original test-fold boundary. Each draw refits preprocessing, selects ridge strength on weighted validation prefixes, refits on non-test texts and reselects the fixed rank rule. The penalty scales with total training-prefix weight, equivalent to duplicating rows; validation ties keep the first strength in descending order. Empty fitting or validation sets would be recorded as failed draws, not replaced. All 499 completed. The unweighted reconstruction preserves every original choice and selected strength.

Table 5 reports both equal-prefix and equal-text contrasts. The central 95% bootstrap ranges include both signs in every condition. Deleting each of the 46 texts and repeating the affected fits gives a separate influence check, not a confidence interval. These results condition on the text panel and fold design; they neither resample human respondents nor account for the earlier choice of conditions to examine. They differ from the finite-sample respondent-pair information bound.

Table 5: Qwen selector minus matched rank rule. Refit ranges are the central 95% of 499 whole-text bootstrap fits. The last column counts positive contrasts after each of 46 single-text deletions.

Train T	Eval T	Weighting	Difference	Refit range	Positive deletions
1	1	Equal prefix	0.0102	$[-0.0151, 0.0385]$	46/46
1	1	Equal text	0.0128	$[-0.0120, 0.0362]$	46/46
1	1.25	Equal prefix	0.0050	$[-0.0121, 0.0251]$	46/46
1	1.25	Equal text	0.0065	$[-0.0101, 0.0237]$	46/46
1.25	1.25	Equal prefix	-0.0040	$[-0.0167, 0.0222]$	23/46
1.25	1.25	Equal text	-0.0013	$[-0.0139, 0.0212]$	27/46

C.6 Selection Targets Before Fitting

We compare mean-score selection with the most frequent per-answer winner using the same 24 support people, candidates and evaluation answers. Repeated answers keep their person multiplicity; missing answers neither vote nor enter means. Exact ties use the first action, with native first. Winner versus mean-score gains are 0.0177 versus 0.0562 nat in OLMoE, 0.0332 versus 0.0689 in Qwen, and 0.0086 versus 0.0353 in Granite. This loss precedes fitting. A separate pointwise sensitivity check resamples support and evaluation people independently within each list, reselects both rules, and retains all eight lists and 4,096 draws per list. Every list's winner-minus-mean 95% percentile interval is below zero; these are not simultaneous intervals or independent model replications.

C.7 Objectives and Nonlinear Scorers

The nonlinear comparison adds a width-128, two-affine-layer residual to the frozen rank ridge, with GELU or identity activation. All 33 decisions retain native at zero. Input weights have standard deviation $1/\sqrt{d}$; biases and readout weights start at zero, so policies initially match ridge. MSE fits utility residuals; direct utility maximizes training utility under a softmax over action scores at temperature 0.1 nat. Both use training-residual RMS to scale predictions, and the direct objective also divides its loss by that RMS. Test selection is deterministic argmax. The shared five Provo folds use $T = 1$ for OLMoE and $T = 1.25$ for Qwen.

Full-batch CUDA FP64 AdamW uses learning rate 0.001, weight decay 0.01, betas $(0.9, 0.999)$, epsilon 10^{-8} and gradient-norm cap five. Validation action gain selects 0, 25, 50, 100, 200 or 400 updates, preferring earlier steps on ties within 10^{-12}; refitting uses that count. The small budget fits 166 prefixes, validates on 64 and refits on 230; the larger budget fits 855–1,016, then re-

fits on all non-test texts. All three seeds (2026091401–2026091403, plus 1000 times fold index) are averaged as policy utilities. Table 6 retains every model/budget/objective combination. OLMoE's small-data direct-utility improvement is driven by one fold; its nonlinear changes there are $-0.0001, -0.0016, +0.0344, 0, 0$ nat.

Table 6: Objective and capacity controls on all 1,581 Provo prefixes. Gains over same-temperature native; three fixed seeds per learned residual.

	Refit	MSE		Direct utility			
Model	prefixes	Affine	GELU	Affine	GELU	Ridge	Rank rule
OLMoE	230	0.0133	0.0133	0.0163	0.0208	0.0133	0.0117
OLMoE	All	0.0382	0.0370	0.0374	0.0368	0.0382	0.0379
Qwen	230	0.0081	0.0092	0.0073	0.0054	0.0089	0.0042
Qwen	All	0.0074	0.0088	0.0053	0.0028	0.0093	0.0133

C.8 Training the Original Gate

We train a copy W of the original expert gate from its complete saved inputs, using the initial W_0 as reference. Let $s_W(a, x)$ sum the logits of a candidate's eight experts, a^- be the current within-bank choice, a^+ the best mean-support choice, and Δ_x their nonnegative utility difference. The equal-prefix loss is

$$-\mathbb{E}_x \Delta_x \log \sigma\big(0.1[s_W(a^+, x) - s_W(a^-, x) - s_{W_0}(a^+, x) + s_{W_0}(a^-, x)]\big).$$

Indices and Δ_x are fixed in the gradient; zero-improvement prefixes stay in the mean. This EPO-style adaptation (Yoon et al., 2026) changes selection within the existing bank, not its expert identities, execution weights or predictions. Initial gates reproduce native expert selection on all UCL and Provo inputs.

Four learning rates ($3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}$) each receive 200 full-batch AdamW updates, with weight decay 0.01 and the betas/epsilon above. Validation selects among their final fits and native; ties prefer native, then the smaller rate. Refit restarts from W_0 for 200 updates on the allowed refit data. UCL uses the original 166/64 split and 230-prefix refit; Provo reuses both budgets and text folds. At $T = 1$, both UCL gates fit a best-mean choice on all 230 training prefixes, but held-out gains in Table 7 remain mixed. Before refitting, OLMoE's UCL gain is positive while its transfer and Provo gains are already negative; the complete pre/refit and rate tables are preserved in the full-results supplement.

Table 7: Original-gate refits and matched controls. Transfer freezes the UCL fit; within-Provo rows hold out whole texts. Ridge selectors use ID-sorted slots or gate ranks; the rank rule is the matched fixed rule.

Model	Setting	Gate	ID-slot ridge	Rank ridge	Rank rule
OLMoE	UCL	-0.002059	0.004646	-0.000693	-0.006649
OLMoE	Transfer	-0.003667	0.001141	0.010063	0.037653
OLMoE	Provo 230	-0.005991	0.004751	0.013293	0.011701
OLMoE	Provo all	-0.005541	0.010090	0.038181	0.037928
Qwen	UCL	0.011361	0.014832	0.009871	0.001079
Qwen	Transfer	-0.004750	0.003228	0.021765	0.022490
Qwen	Provo 230	0.001991	0.002818	0.022334	0.010677
Qwen	Provo all	0.000247	0.006628	0.031525	0.021317

Gains in nats; native = 0.

C.9 Complete Expert Outputs

Four OLMoE arms add the same width-128 GELU candidate scorer to rank ridge. All receive the complete pre-MoE state, gate logits, candidate weight and membership changes, and added/removed gate ranks. Full output features add the candidate's 2,048-dimensional weighted-output difference from native, its norm and cosine with the expert input. Compression projects the difference onto 64 fixed orthonormal directions and back, retaining norm and cosine. Scrambling deranges output

features across the 32 non-native actions at each prefix. The no-output arm zero-fills this block; it matches allocated parameters, not the effective hypothesis class. Context, candidate and output linear maps are summed before activation and scalar readout; native subtraction and zero-initialized readout start every arm at ridge.

All arms retain the five Provo folds, all-data ridge fits and three seeds, with train-only feature normalization and training-residual RMS scaling. FP32 AdamW uses learning rate 0.001, weight decay 0.01, gradient cap five and batches of 64 prefixes with all actions. Initial weights and batch schedules are paired within seed/fold. Validation hard-action gain selects the same 0–400 update grid, with earlier ties, then refits for that count; saved parameters are evaluated in CUDA FP64. Full, compressed and scrambled arms execute twelve experts before choosing eight; compression does not reduce acquisition cost. Outputs use the complete weighted team, rescaled to native selected mass, not just added-minus-removed expert outputs.

Table 8 shows that the full-output deficit exists before refitting. Full-output policies beat their own ridge on training prefixes by 0.0262 nat before refit and 0.0248 afterward, but lose on held-out texts in both cases. Full also loses to no-output features in four of five folds; the only positive fold adds 0.0003 nat. Equal-text weighting preserves the aggregate ordering. A realizable generated-target check verified the learning interface before human-data fitting, and parameter-based replay reconstructs all selected actions. The full seed/fold table remains available without requiring it to interpret this matched comparison.

Table 8: Expert-output controls before and after refitting. Mean gains on the same 1,581 Provo test prefixes at $T = 1$, averaging three seeds. Each excess uses its own ridge (0.0381 before, 0.0382 after).

	Test gain		Excess over own ridge	
Inputs	Before refit	After refit	Before refit	After refit
Pre-execution	0.0360	0.0329	-0.0021	-0.0053
Compressed outputs	0.0222	0.0228	-0.0159	-0.0154
Full outputs	0.0266	0.0275	-0.0115	-0.0107
Scrambled outputs	0.0365	0.0371	-0.0016	-0.0010

D A KNOWN REFERENCE DISTRIBUTION

The human experiments bound information value because the population of possible answers is unknown. Here a fixed reference model supplies that population, so both expectations in Equation 1 can be evaluated over the vocabulary. We keep OLMoE-1B-7B-0924 frozen and use OLMoE-1B-7B-0125-Instruct at temperature one as the reference, with the same raw 128-token prefix and no chat template. Native plus 32 one-expert swaps at layer 15 form the candidate set. The reference is fixed before the engineering pilot, not selected for larger routing gains. It defines a controlled prediction problem, not a model of the human respondents.

Data and scoring. One local FineWeb-Edu shard supplies document-disjoint engineering, training, validation and test sets. Exact duplicate documents and earlier project samples are excluded before outcome calculation. Of the first 1,000 and 10,000 training documents, 994 and 9,893 have at least 129 tokens; the larger set has 9,886 distinct prefixes. Validation has 990 eligible documents. The 2,000-document test manifest yields 1,969 eligible documents; one exact prefix overlaps development, leaving 1,968 distinct new prefixes for the primary analysis. No excluded document is replaced. We score the next original token, not a case-pooled word event. Token strings align the two checkpoint vocabularies, including their two remapped IDs. Reference probabilities are normalized over shared tokens; mean excluded mass on test is 7.8×10^{-9}. Both models use FP32, with TF32 disabled; expectations and statistical calculations use CUDA FP64.

Learning and selection. The primary target is each candidate's expected gain under the complete reference distribution. The secondary target is its gain on the single observed corpus token. Context features contain the 2,048-dimensional normalized pre-MLP state and 64 centered gate logits. The 140 candidate features encode differences in expert membership and weights, plus removal and addition ranks. A 2,050-dimensional output block contains the full routed-output difference, its norm

and its cosine similarity with the normalized pre-MLP state. The expert-output MLP receives this block, executing twelve experts before selecting eight; the State MLP retains its parameters but supplies zeros. Residual variants append the 2,048-dimensional pre-MLP residual and therefore have more parameters. Feature means and standard deviations are fitted on training data only, pooling all 33 actions for candidate features and flooring standard deviations at 10^{-5}.

The candidate MLP shares one scoring function $f_\theta(x, a)$ across all actions: linear context and candidate contributions are summed, followed by one width-128 GELU hidden layer and a scalar linear output. We anchor scores at native, $s_\theta(x, a) = f_\theta(x, a) - f_\theta(x, a_0)$. For a batch B, the loss is

$$\frac{1}{32|B|} \sum_{i \in B} \sum_{a \neq a_0} \big(s_\theta(x_i, a) - t_{ia}/r\big)^2,$$

where t_{ia} is the training-target gain and r is its training RMS over all non-native actions, floored at 10^{-8}, without target centering. Thus all 32 alternatives have equal weight and native is exactly zero. The chosen action is $\arg\max_a s_\theta(x, a)$ over all 33 candidates; ties use the first candidate, with native first. Scores are multiplied by r when reporting predicted gains.

Training samples batches of 64 documents uniformly with replacement. We use Adam with learning rate 0.001, standard betas $(0.9, 0.999)$, epsilon 10^{-8}, no weight decay and gradient-norm clipping at five. Input weights are Gaussian with inverse-square-root total feature-dimension scaling; hidden bias and output weights start at zero. Seeds 20260925–20260927 are retained separately. Validation decision gain, not regression loss, selects between 200 and 800 updates per seed, keeping the earlier checkpoint on ties. Ridge is a separate context-only linear baseline with an unpenalized intercept and penalties 0.01, 0.1 and 1; validation selects its penalty. No seed is selected.

Using the output layer. A further matched pair receives the residual-augmented context, each candidate's final normalized state change and its expected gain under native's own token distribution. Let z_a be that final state, W the output matrix, and p_a the candidate prediction. The linear output layer gives the identity

$$\mathbb{E}_q g(a, Y) = -\mathrm{KL}(p_0 \| p_a) + \big(\mathbb{E}_q W_Y - \mathbb{E}_{p_0} W_Y\big)^\top (z_a - z_0).$$

The generic readout MLP predicts gains from these inputs. The structured version instead learns one context-dependent vector and takes its inner product with each state change, then adds the native-distribution penalty. Both use expected-gain labels; neither receives reference probabilities at prediction time. State changes use per-coordinate training RMS scaling without centering; labels and the penalty use target-RMS units. The structured inner product is scaled by $1/\sqrt{2048}$, and both scorers subtract their native score before the same squared-error training and argmax decision. This contrast makes the native penalty additive and constrains the learned state interaction. It requires twelve expert outputs and 33 output-head evaluations, making it an expensive computation control rather than a cheap pre-execution router.

Development trajectory. Longer fitting increased training gain without improving validation in the larger original State MLP runs. For the same three seeds trained on 9,893 documents, mean expected training gain rose from 0.0075 nat at 200 updates to 0.0377 at 800; mean validation gain fell from -0.0057 to -0.0129. All three selected checkpoints were at 200 updates. These saved development records compare the original State MLP, not the residual variant or new test policies. They explain the validation choice within the tested schedule; fixed update counts and fewer data passes at the larger budget do not establish convergence or rule out other training procedures.

Frozen evaluation and uncertainty. The residual and readout comparisons were added after negative generic-MLP validation results. All families, data budgets, seeds and validation-selected checkpoints were then frozen before test access: 60 policies in total, including the observed-token-trained controls. Gains first average the separately fitted seeds' utilities, not their predictions. Approximate 95% intervals use 5,000 paired bootstrap draws of exact-prefix clusters, retaining document weights and conditioning on the fitted policies. All 33 actions are checked against full forwards on the first two test prefixes, with periodic checks thereafter. An independent readout check recomputes every test row; the maximum expected-gain discrepancy is 2.7×10^{-15}. These checks preserve the original candidate predictions.

Table 9: Known-population study: expected-gain supervision. Train and test columns use the same frozen, validation-selected policies and report gain over native in nats. MLP rows average three seeds; intervals condition on the fitted policies. Budgets 1k/10k contain 994/9,893 documents.

Training	Selector	Train	Test	95% test interval
1k	Native	0.0000	0.0000	$[0.0000, 0.0000]$
1k	Fixed rank rule	0.0000	0.0000	$[0.0000, 0.0000]$
1k	Ridge	0.0415	-0.0161	$[-0.0199, -0.0124]$
1k	State MLP	0.0463	-0.0119	$[-0.0189, -0.0072]$
1k	Expert-output MLP	0.0449	-0.0095	$[-0.0123, -0.0070]$
1k	State MLP + residual	0.0459	-0.0139	$[-0.0211, -0.0089]$
1k	Expert-output MLP + residual	0.0495	-0.0084	$[-0.0111, -0.0060]$
1k	Readout MLP	0.0465	-0.0064	$[-0.0080, -0.0048]$
1k	Structured readout	0.0401	-0.0051	$[-0.0062, -0.0041]$
10k	Native	0.0000	0.0000	$[0.0000, 0.0000]$
10k	Fixed rank rule	0.0000	0.0000	$[0.0000, 0.0000]$
10k	Ridge	0.0158	-0.0090	$[-0.0119, -0.0061]$
10k	State MLP	0.0075	-0.0050	$[-0.0069, -0.0030]$
10k	Expert-output MLP	0.0163	-0.0087	$[-0.0107, -0.0067]$
10k	State MLP + residual	0.0064	-0.0055	$[-0.0075, -0.0035]$
10k	Expert-output MLP + residual	0.0250	-0.0118	$[-0.0141, -0.0096]$
10k	Readout MLP	0.0263	-0.0098	$[-0.0115, -0.0081]$
10k	Structured readout	0.0117	-0.0081	$[-0.0095, -0.0067]$

What the comparison establishes. The known input-only optimum is 0.0571 nat on the primary test set, with 95% interval $[0.0531, 0.0613]$. All expected-target learned family means in Table 9 are below native, as are their reported intervals. The same learners also have negative mean gains on the actual corpus tokens. The reference model itself scores those observed tokens 0.3760 nat below native on the 1,969 eligible test documents. Thus the reference optimum is not a demonstrated improvement in real-text prediction; it supplies a known target against which to measure recovery. The result identifies value missed by these feature–scorer pairs, not the best value learnable from the complete prefix.

Table 10: Observed-token-supervised controls, evaluated under the same reference distribution. Checkpoints retain their original observed-token validation selection, rather than being reselected against reference gains. Train and test columns therefore measure a different target from supervision.

Training	Selector	Train	Test	95% test interval
1k	Native	0.0000	0.0000	$[0.0000, 0.0000]$
1k	Fixed rank rule	0.0000	0.0000	$[0.0000, 0.0000]$
1k	Ridge	-0.0227	-0.0200	$[-0.0245, -0.0155]$
1k	State MLP	-0.0278	-0.0143	$[-0.0183, -0.0105]$
1k	Expert-output MLP	-0.0268	-0.0209	$[-0.0250, -0.0171]$
10k	Native	0.0000	0.0000	$[0.0000, 0.0000]$
10k	Fixed rank rule	0.0000	0.0000	$[0.0000, 0.0000]$
10k	Ridge	-0.0189	-0.0217	$[-0.0263, -0.0175]$
10k	State MLP	-0.0165	-0.0115	$[-0.0141, -0.0091]$
10k	Expert-output MLP	-0.0225	-0.0183	$[-0.0214, -0.0153]$

Table 10 retains the secondary supervision comparison. The accompanying `supplement/known_optimum_policies.csv` lists every frozen seed and checkpoint, training/validation gains, test gains under both targets and expected rank-action shuffle gains. No additional model, teacher, threshold or seed was selected using these test results.

E Expert Operations and Score Controls

E.1 Local Identity and Weight Controls

On OLMoE's 1,581 Provo prefixes, all arms retain eight experts, native total mass and 33 indices, including native at zero. Identity-only gives the replacement expert the removed expert's native weight. Weight-only retains native identities but uses joint-swap weights, assigning the replacement's weight to the removed expert. Joint is the original renormalized swap; scores are case-pooled at $T = 1$.

At each prefix's joint-best-mean index, identities alone gain 0.1704 nat, above the joint gain of 0.1467. The UCL-frozen rule (remove highest-gate native, add third-highest unselected) reverses this ordering: joint gains 0.0377 versus identity-only 0.0350 and weight-only 0.0373. Equal-text weighting retains the reversal. In both conditions joint gain is below the sum of the single-change gains; they are not additive shares. Answer-selected values fit and evaluate the same histogram. The frozen-rule diagnostic also follows inspected results, so these rankings remain descriptive.

E.2 Complete-Team Assignments

An answer-blind shuffle within eight UCL lists forms 119 pools of at most eight prefixes, keeping all 917 prefixes and remainders. Each donor supplies its native eight expert IDs and applied weights; recipients compute those experts on their own states. Only final-token final-layer routing changes. Reuse permits repeated donors; once-only requires a permutation, preserving physical expert calls and the ID/weight-pair inventory. Unlike local swaps, donor weights are not renormalized to recipient mass. This fixed offline bank is not an autoregressive schedule or a constraint on search cost.

Assignments optimize each person's answer vector, the shared evaluation mean, or the original 24 support people's mean before separate-person evaluation. Missing answers have zero utility for every assignment and stay in the fixed prefix denominator; pool means are prefix-count weighted. Table 11 shows support benefit but negative uniform reassignment. Shared-evaluation choices are optimistically fitted benchmarks, not population common values; these are descriptive same-panel comparisons, not new-text tests or confidence bounds.

Table 11: Complete-team case-pooled gains over native at $T = 1$ (nats).

	OLMoE		Qwen	
Choice	Reuse	Once	Reuse	Once
Per person	0.4171	0.3565	0.3072	0.2604
Shared evaluation	0.1418	0.1194	0.1053	0.0824
Support → evaluation	0.1098	0.0949	0.0818	0.0626
Uniform permutation	-0.0550	-0.0550	-0.0742	-0.0742

E.3 Common Assignments and Separate Search

For OLMoE, common physical IDs align with themselves; remaining donor and recipient IDs pair in ascending order. Identity-only combines donor IDs with aligned recipient weights; weights-only combines recipient IDs with aligned donor weights. Joint uses both donor components, without renormalization. Once-only preserves expert call counts in every arm, but only native and joint generally preserve the ID/weight-pair inventory.

Table 12 holds the joint-selected assignment fixed or optimizes each arm separately, retaining every person and candidate. Once-only support-selected reweighting improves from 0.0152 to 0.0445 nat; joint leads per-person optima, not support means. These banks are nonnested: a joint donor fixes one identity/weight combination, not every possible combination. Their different masses and effective action counts preclude interpreting differences between own optima as additive effects.

Execution checks. Forced-forward checks cover selected cells, not every donor assignment. One OLMoE batch failed the 10^{-5}-nat threshold; the complete panel was rerun without omissions. Own-search checks recover saved choices within 2×10^{-5}. Local Provo checks cover all 33 actions; three ambiguous residual scales required fresh FP32 forwards, with no prefix exclusions. Counts and failures are in `supplement/full_results_v17.pdf`, pp. 34–39, 43.

Table 12: OLMoE gains (nats): identity (I), weights (W), joint (J).

	Joint-selected assignment			Each arm's own assignment		
Constraint: selection	I	W	J	I	W	J
Once: per person	0.2937	0.0791	0.3565	0.3414	0.1750	0.3565
Once: shared evaluation	0.0984	0.0212	0.1194	0.1233	0.0585	0.1194
Once: support	0.0759	0.0152	0.0949	0.0967	0.0445	0.0949
Reuse: per person	0.3299	0.1026	0.4171	0.3890	0.2154	0.4171
Reuse: shared evaluation	0.1151	0.0235	0.1418	0.1415	0.0729	0.1418
Reuse: support	0.0909	0.0157	0.1098	0.1126	0.0552	0.1098

E.4 Scoring Horizon and Scope

The full-word check changes OLMoE's ninth-layer final prefix position; seven downstream blocks and teacher-forced response tokens execute natively. All 917 prefixes/33 actions remain; 9,197 of 72,214 responses are multi-token. Score the original-case word's fixed token sequence by summed full-vocabulary log probabilities, without terminator, length adjustment or word renormalization: this is finite-response utility, not a normalized lexical distribution. Holding first-token support choices fixed gives $0.053416 + 0.000167 = 0.053583$ nat; full-word reselection changes 11.56% of choices but gains only $0.052882 + 0.000954 = 0.053836$. Original middle-layer raw/class Hoeffding bounds are zero; the sequence study's first-token/full-word Hoeffding bounds also remain zero, while CDF bounds are 0.015386/0.015857 (joint 95%, $0.05/32$ per list/outcome/method, 0.1-nat cap after pair averaging). Final-layer intervention leaves forced suffix scores exactly unchanged.

Of 5,568 teacher-forcing checks, one prefix required a disclosed singleton-layout amendment; all 33 actions/60 sequences then passed the unchanged 10^{-5} gate, with no exclusions. Detailed checks are archived on pp. 40–41.

F Research Coverage and Scope

The original human-policy predictions preceded their test scoring. Later coordinate, calibration, capacity, objective and expert-operation comparisons reused inspected human panels; they refine interpretation rather than provide new independent confirmation. Earlier single-response FineWeb evaluation on 8,192 new documents found 0.192621 nat of hindsight gain but -0.003410 global and -0.004505 conditional gain. Compressed-output and external-critic probes were negative; action-aligned and readout probes were closed after development results. These branches remain in the full-results archive.

The known-reference study uses a separate frozen test manifest. Residual-state and readout variants followed negative development results; all 60 policies were locked before test access (Appendix D). The proposed 100k expansion was not run. The portable supplement preserves every seed-level record and the earlier full manuscript, indexed by topic.

Numerical checks and failures. Replay checks use a 10^{-5}-nat full-forward tolerance. Every Provo prefix checks native, selected policies/rules and a cyclic alternative; anchors cover complete banks. A Qwen cache mismatch required batch-preserving replay; OLMoE panel and suffix reruns are described in Appendix E. No tolerance or inclusion rule changed. Saved-parameter reconstruction separately checks learned choices. Aggregation and statistics use CUDA FP64; fitting precision follows each learner's protocol. These are correctness, not latency, checks.

Scope. These short English log-score tests do not measure semantics, reasoning or long-context behavior. The depth check covers one OLMoE location. Fixed active count need not balance physical loads: Qwen's load total variation from native is 0.0897, with one formerly used expert unvisited. Finite learners and budgets cannot establish intrinsic unlearnability or fully separate learning from computation limits.