NATIVE ROUTING PRIORS FOR BUDGETED LOW-RANK ADAPTATION OF MIXTURE-OF-EXPERTS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Under a low-rank parameter budget, adapting a mixture-of-experts (MoE) model couples expert coverage, local rank and expert identity. We examine these decisions through Native Router-Guided Layer–Expert Capacity Allocation (NRG), using reference-answer-free profiles and static DoRA adapters. On Qwen3.5-35B-A3B-Base, selected rank-32 adapters exceed approximately budget-matched all-expert adapters by 0.83–0.90 percentage points across three low-data VCR and NLVR2 settings. Grouped intervals support a positive conditional difference on NLVR2, while VCR intervals cross zero. Follow-up controls reveal that expert selection alone does not ensure favorable accuracy: on VCR Train50, selected rank-7 adapters fall 0.90 points below all-expert rank-7 adapters, while increasing rank on the same selected experts yields a 1.77-point gain. The corresponding NLVR2 rank comparison remains uncertain, and neither task establishes an advantage for routing-derived over uniform layer quotas. Historical execution compatibility is only partially verified, limiting attribution in these follow-up comparisons. Repeated random allocations retain a favorable VCR mean difference, but an exclusion-based probe does not identify a ranking mechanism. These observations motivate separate tests of capacity distribution and expert identity when evaluating routing priors. We provide budget accounting and numerical reproduction; the evidence remains conditional on observed models from one backbone and reused development subsets.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) models route each token to a subset of their experts (Fedus et al., 2022). Parameter-efficient adaptation introduces a second allocation decision: which of those experts should receive trainable updates? The two decisions operate at different levels. The router selects computation for an input token, whereas adapter placement determines which expert parameters can change during training. Under a fixed low-rank budget, adapting more experts leaves less capacity for each expert. MoE adaptation therefore couples expert coverage, local rank and expert identity.

Native routing statistics provide an accessible prior for this allocation: they measure how the pretrained model distributes routing mass over calibration inputs. Expert-specialized fine-tuning (ESFT) uses task-dependent scores and cumulative per-layer selection (Wang et al., 2024); MoE-Sieve combines routing-selected experts with LoRA (Manzoni, 2026). Building on this family of methods, we ask: *under a low-rank budget, what do routing-guided expert selections improve, and which comparisons support that conclusion?* The question concerns adaptation of the MoE expert bank; multimodal reasoning supplies the evaluation setting in this study.

An all-expert baseline and a random-subset baseline answer different questions. Comparing selected experts at higher rank with all experts at lower rank tests a change in capacity distribution. Comparing routing-ranked and random subsets at identical per-layer counts and ranks instead tests expert identity. Repeating the random selection then measures sensitivity to the sampled expert list. This distinction is essential when interpreting a favorable routing-guided result: an advantage over all-expert adaptation can coexist with uncertain evidence for the ranking itself.

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.

We instantiate NRG with DoRA on Qwen3.5-35B-A3B-Base. Routing-selected rank-32 adapters have the highest three-seed means in the original four-arm comparisons. Follow-up VCR results sharpen the interpretation: the selected subset falls below all-expert adaptation at the same low rank, although raising rank on that subset improves its mean. Thus a favorable joint configuration does not establish that expert concentration alone is beneficial. Neither task resolves a layer-quota advantage, and historical execution gaps limit follow-up attribution. Our contributions are:

- Evidence that routing-selected high-rank DoRA has higher observed means than approximately budget-matched all-expert DoRA in three settings, with grouped conditional interval support on NLVR2.
- A task-dependent contrast between favorable high-rank selection and unfavorable same-low-rank selection on VCR, alongside unresolved layer-quota benefits; these results constrain attribution to the routing prior.
- Repeated-allocation and exclusion controls, together with explicit budget accounting, that distinguish observed configuration gains from a general expert-ranking claim.

The contribution is empirical: the controlled comparisons clarify what can be inferred from native routing priors under a low-rank budget. The budget analysis applies to compatible expert banks of different sizes; the experiments evaluate one MoE backbone, so transfer of the accuracy gains remains an open question.

2 Related work

Low-rank adaptation and capacity allocation. LoRA parameterizes weight updates through low-rank factors, and DoRA adds magnitude adaptation (Hu et al., 2021; Liu et al., 2024). AdaLoRA allocates a matrix-level rank budget during optimization (Zhang et al., 2023). In an MoE expert bank, adapter capacity also depends on which layer–expert slots receive updates. NRG uses the established DoRA parameterization and fixes expert membership before supervised training, enabling comparisons at a common rank and per-layer expert count.

Routing-informed expert adaptation. ESFT provides the principal basis for routing-derived relevance, cumulative layer selection and random controls with matched layer counts (Wang et al., 2024). MoE-Sieve is a close precedent for static routing-guided LoRA and matched-budget random comparisons (Manzoni, 2026). NRG inherits these selection ideas. Its empirical focus is their behavior with DoRA under joint coverage/rank changes, layer-quota and fixed-membership rank controls, replicated random allocations and an exclusion-based probe. Dynamic alternatives include routing- and gradient-informed rank growth in DR-LoRA and routing-conditioned adapters in MoE2-LoRA (Deng et al., 2026; Yang et al., 2026). Appendix A compares the designs; these methods have not been evaluated against NRG under a shared protocol.

Expert adaptation and expert removal. Infrequent activation need not imply unimportant knowledge (He et al., 2026). Our BOTTOM arm places adapters on low-mass experts while retaining the entire pretrained expert bank. Its outcome informs where to place updates; it does not measure whether experts can be pruned.

3 Budgeted adapter allocation in MoE models

We treat a layer–expert pair as an adapter allocation unit. NRG first summarizes native routing on calibration inputs, then fixes the number and identities of adapted experts, and finally trains their low-rank updates. Membership is static during training; all pretrained experts remain available for forward routing. Throughout the paper, *Dense* means adapters on every routed expert in the same MoE backbone, not a dense-architecture model or full-parameter fine-tuning. Figure 1 shows the allocation and training stages.

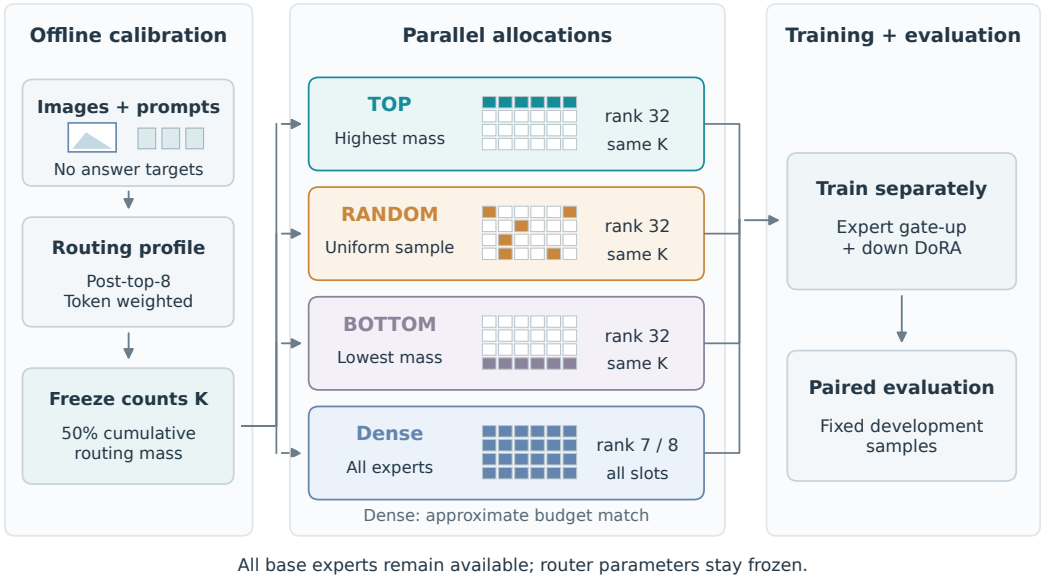

Figure 1: Original four-arm MoE adapter allocation under a low-rank budget. Native routing informs static adapter placement. Layer counts are fixed before supervised adaptation. Expert-grid icons illustrate allocation patterns, not measured expert identities or counts. TOP, RANDOM and BOTTOM retain the same counts and rank; Dense uses all routed experts at a lower integer rank (7 for VCR, 8 for NLVR2). All base experts remain available to forward routing. Frozen router parameters do not imply unchanged token-wise routes after expert adaptation.

3.1 Reference-answer-free routing profiles

The evaluated model has a language-model component reported as 35B total parameters and 3B activated parameters, with 40 MoE layers, 256 routed experts per layer and native top-eight routing with a shared expert (Qwen Team, 2026). Total and activated parameter counts describe different aspects of this sparse architecture. Router, attention, shared-expert, vision-encoder and projector parameters remain frozen. Updates target routed-expert gate-up and down projections.

Let p_{lte} be the native softmax routing probability for expert e at token t in layer l, and $\mathcal{T}_{lt}$ the native top-eight set. We aggregate actual post-selection, renormalized routing contributions:

$$w_{lte} = \frac{p_{lte}\mathbf{1}[e \in \mathcal{T}_{lt}]}{\sum_{j\in\mathcal{T}_{lt}} p_{ltj}}, \qquad q_{le} = \frac{\sum_{t\in\mathcal{C}} w_{lte}}{\sum_{j=1}^{E}\sum_{t\in\mathcal{C}} w_{ltj}}. \tag{1}$$

Here $\mathcal{C}$ indexes calibration tokens. Profiling uses image and user-prompt tokens, including template special tokens, without an assistant target, loss or backward pass. VCR answer choices remain in the prompt; the correct answer label is excluded. The aggregate is token-weighted, so longer inputs contribute more mass. The profile is computed before adaptation and then frozen. Appendix I gives allocation pseudocode and the interface to the recorded expert lists.

3.2 Layer coverage and expert identity

Sort q_{le} within each layer in descending order. A cumulative threshold $\tau = 0.5$ determines

$$K_l = \min\left\{k : \sum_{j=1}^{k} q_{l,(j)} \geq \tau\right\}. \tag{2}$$

TOP selects the K_l highest-mass experts; BOTTOM selects the K_l lowest-mass experts; RANDOM samples K_l distinct experts from the whole layer. Frozen manifests specify exact identities. The

Table 1: Original four-arm adapter budgets. Sparse denotes TOP, RANDOM or BOTTOM at rank 32. Mismatch is $(P_{\text{Dense}} - P_{\text{Sparse}})/P_{\text{Sparse}}$.

Task	Sparse slots	Sparse r	Sparse params	Dense r/params	Mismatch
VCR	2,405	32	440,826,880	7 / 435,159,040	-1.286%
NLVR2	2,549	32	467,221,504	8 / 492,830,720	$+5.481\%$

same lists are reused across the relevant training seeds. Thus these three arms control layer counts, adapter rank and parameter count while changing membership. K_l is the number of adapted experts, not a replacement for native top-eight forward routing. We do not optimize it against development accuracy.

3.3 Trading expert coverage for local rank

For selected matrices, LoRA represents $W' = W_0 + (\alpha/r)BA$; DoRA additionally learns output-channel magnitudes (Liu et al., 2024). With hidden width 2,048 and expert width 512, the fused gate-up and down factors contain $5632r$ parameters per expert, and DoRA magnitudes add 3,072. Consequently,

$$P_{\text{DoRA}} = \sum_{l=1}^{40} K_l(5632r + 3072). \tag{3}$$

TOP, RANDOM and BOTTOM in the original four-arm experiments use $r = 32$, $\alpha = 64$; follow-up ranks are specified in Appendix H. Dense adapts all 10,240 layer–expert slots at the nearest uniform integer rank, with $\alpha = 2r$. Table 1 records the resulting mismatch. Counts measure trainable adapters, not activation memory, FLOPs or training time.

3.4 Budget accounting beyond the evaluated backbone

For L layers with hidden width h_l, expert width d_l, selected count K_l and rank r_l, fused gate-up and down DoRA adapters require

$$P = \sum_{l=1}^{L} K_l(c_l r_l + m_l), \qquad c_l = 2h_l + 3d_l, \quad m_l = h_l + 2d_l. \tag{4}$$

The coefficients count the low-rank factors and output-channel magnitudes. This expression applies to larger compatible expert banks and makes integer-rank mismatches explicit. It describes parameter allocation, not expected accuracy or system efficiency. Appendix F derives the corresponding coverage/rank relation and width-scaling behavior. Each new backbone requires its own routing profile and budget calibration.

4 Experimental design

Questions and controls. We ask about (Q1) joint expert coverage/rank allocation, (Q2) expert identity at fixed layer counts and rank, (Q3) exclusion of low-mass experts, and (Q4) layer quotas and local rank. Table 2 separates controlled and changing variables. RANDOM samples experts uniformly without replacement; its repeated allocations are called *Uniform-random*. *Uniform-layer TOP* instead redistributes the same total slots evenly across layers and remains routing-ranked within each layer. The follow-up designs and execution limitations are detailed in Appendix H.

Tasks and reuse. We use VCR image-and-question-to-answer classification with four alternatives (Zellers et al., 2019) and NLVR2 true/false reasoning over two photographs (Suhr et al., 2019). Each frozen development subset contains 1,000 examples. NLVR2 comprises 250 four-example statement groups within 226 image sets and balanced true/false labels. These are development results, not official test scores. VCR Train50 is a subset of Train100; both reuse Train100 inputs for reference-answer-free calibration. Thus calibration and supervised inputs overlap, with labels excluded during profiling. NLVR2 uses 100 supervised and 100 calibration inputs. Each public

Table 2: Comparison design. K denotes the vector of per-layer adapted-expert counts; $S = \sum_l K_l$. All comparisons share the intended backbone and task-specific data. Follow-up comparisons (†) have partially verified historical/new execution compatibility.

Comparison	Fixed allocation variables	Changed allocation variables	Budget
TOP32 vs Dense	—	Coverage, identities, rank	Approx.
TOP32 vs RANDOM/BOTTOM	K, rank	Identities	Equal
NonBottom vs Uniform-random	K, rank	Eligible pool, identities	Equal
TOP32 vs Uniform-layer TOP32†	S, rank	Layer quotas, identities	Equal
TOP32 vs TOP-low-rank†	K, identities	Rank, parameter count	Unequal
TOP-low-rank vs Dense†	Rank	Coverage, identities, parameter count	Unequal

four-arm comparison has three training seeds. One earlier seed is reused for VCR Train100 and NLVR2. Repeated analyses do not create new independent evaluation sets. We make no claim about pretraining contamination.

Training and scoring. Public runs use three epochs, learning rate 5×10^{-5}, cosine scheduling, warmup ratio 0.03, bfloat16 and gradient checkpointing. The sequence cutoff is 3,072; per-device batch size and accumulation are both one. The original 63 recorded runs use eight processes and logged global batch eight, with 39 updates for Train100 and 21 for Train50. Their comparison therefore changes both supervised data and update count. The 12 follow-up runs use the same stated training recipe, with three matched seed identities per task; historical/new execution equivalence is only partially verified. For the original cohort, runtime checks confirm the routed-expert optimizer totals and matching learning-rate histories within each training size; Appendix B describes parameter rebinding and the limits of reconstructing individual sample exposure. VCR accepts only an exact A/B/C/D answer after whitespace trimming and uppercasing; NLVR2 accepts only true/false after trimming and lowercasing. Invalid outputs count as incorrect. Deterministic decoding allows at most 16 and eight answer tokens, respectively. Appendix B provides settings and seed identities.

Estimation and uncertainty. We report direct paired correctness differences and distinguish seed SD from confidence intervals (CIs) and percentage points (pp) from relative percentages. Procedures must match the evaluation design (Dror et al., 2018). Our post-hoc reanalysis conditions on the observed trained models, training data, routing profiles and expert lists. Its target is the example-weighted accuracy difference under resampling of evaluation groups, not performance averaged over new training seeds or new allocations. We first average each item's paired contrast across observed models, then resample whole VCR movies (181 groups) or NLVR2 statements (250 groups), using common weights across arms. A coarser NLVR2 image-set analysis and shared-item resampling assess sensitivity. Group exchangeability and independence across the chosen groups remain assumptions.

All current intervals use 20,000 grouped percentile replicates and condition on the observed models. Table 7 indexes the comparison families, interval levels and sampling units; none is a whole-paper correction. The three primary TOP–Dense contrasts use 98.333% intervals, the nine plotted contrasts use 99.444%, and the four follow-up primary contrasts use 98.75%. P7 components use 97.5%; their total has a descriptive 95% interval. Common group weights preserve dependence from reused TOP predictions. These levels do not guarantee exact finite-sample coverage. Historical deviations and RNG/quantile conventions remain in Appendices C and H; the conditional reanalysis is post-hoc.

Table 3: Accuracy (%) as mean ± sample SD over three training seeds on the same frozen development subset. Bold marks the largest observed mean, not a significance test.

Setting	TOP	RANDOM	Dense	BOTTOM
VCR Train100	$\mathbf{77.867} \pm 0.058$	77.300 ± 0.173	77.033 ± 0.153	76.067 ± 0.404
VCR Train50	$\mathbf{77.233} \pm 0.208$	76.267 ± 0.208	76.367 ± 0.058	75.667 ± 0.058
NLVR2 Train100	$\mathbf{93.467} \pm 0.379$	92.933 ± 0.643	92.567 ± 0.643	92.300 ± 0.529

5 Results and analysis

5.1 Selected high-rank adapters improve means over all-expert adapters

TOP has the highest three-seed mean in all three settings (Table 3). Its observed TOP–Dense differences are 0.833, 0.867 and 0.900 pp. Grouped 98.333% intervals are $[-0.150, 1.885]$ for VCR Train100, $[-0.239, 2.083]$ for VCR Train50 and $[0.133, 1.733]$ for NLVR2 Train100. The latter remains positive with coarser image-set grouping, $[0.136, 1.762]$. Thus the mean ordering is consistent, while conditional interval support differs across tasks. For Q1, this supports a favorable observed coverage/rank configuration; it leaves their individual contributions unresolved.

TOP also exceeds BOTTOM in observed means by 1.800, 1.567 and 1.167 pp. Under the nine-contrast correction in Figure 2, grouped intervals exclude zero for VCR Train100 and NLVR2, but not VCR Train50. This is evidence about the tested adapter memberships, with all pretrained experts retained for routing.

The original RANDOM comparison yields differences of $+0.567$, $+0.967$ and $+0.533$ pp; all three grouped 99.444% intervals include zero. TOP is not higher in every NLVR2 seed: seed2 RANDOM reaches 93.4% versus TOP's 93.3%. Favorable mean ordering alone therefore does not establish a general ranking advantage.

5.2 Follow-up controls separate layer quotas from local rank

We add 12 runs across VCR Train50 and NLVR2 Train100: Uniform-layer TOP32 and TOP-low-rank, each with three seeds. Historical TOP32 and Dense also contribute three models per task. The uniform-layer arm preserves total slots, rank and parameter count, but redistributes slots across layers; its within-layer selection remains routing-ranked. The low-rank arm instead preserves the original TOP expert list and lowers rank to 7 (VCR) or 8 (NLVR2), keeping $\alpha/r = 2$.

TOP32 minus Uniform-layer TOP32 is $+0.367$ pp on VCR and $+0.233$ pp on NLVR2; both 98.75% intervals cross zero (Figure 3). Thus the experiment does not establish an independent benefit from routing-derived layer quotas. It also does not establish equivalence: effects within the reported intervals remain compatible with the data.

On the original TOP expert list, rank 32 exceeds rank 7 by $+1.767$ pp on VCR, with interval $[0.459, 3.513]$. Each leave-one-seed-out difference remains positive ($+1.550$ to $+1.900$ pp). The NLVR2 rank-32 versus rank-8 difference is $+0.367$ pp with interval $[-0.633, 1.367]$; image-set resampling gives the same qualitative conclusion. This supports a favorable VCR higher-rank configuration conditional on the observed models, without establishing a task-general rank benefit. Lowering rank also reduces trainable capacity, so this comparison is not budget-matched.

The secondary same-rank TOP–Dense comparison is -0.900 pp on VCR and $+0.533$ pp on NLVR2. In VCR, the favorable TOP32–Dense7 difference therefore coexists with an unfavorable TOP7–Dense7 difference. This ordering ties the interpretation to the tested rank/coverage configuration: the same-rank comparison changes coverage and parameter count, and the fixed-expert comparison changes rank and parameter count. Neither is a causal decomposition of the joint gain. Complete seed results, intervals and budgets are in Appendix H. These comparisons remain subject to the historical execution gaps summarized in Appendix H; interval exclusion does not remove that qualification.

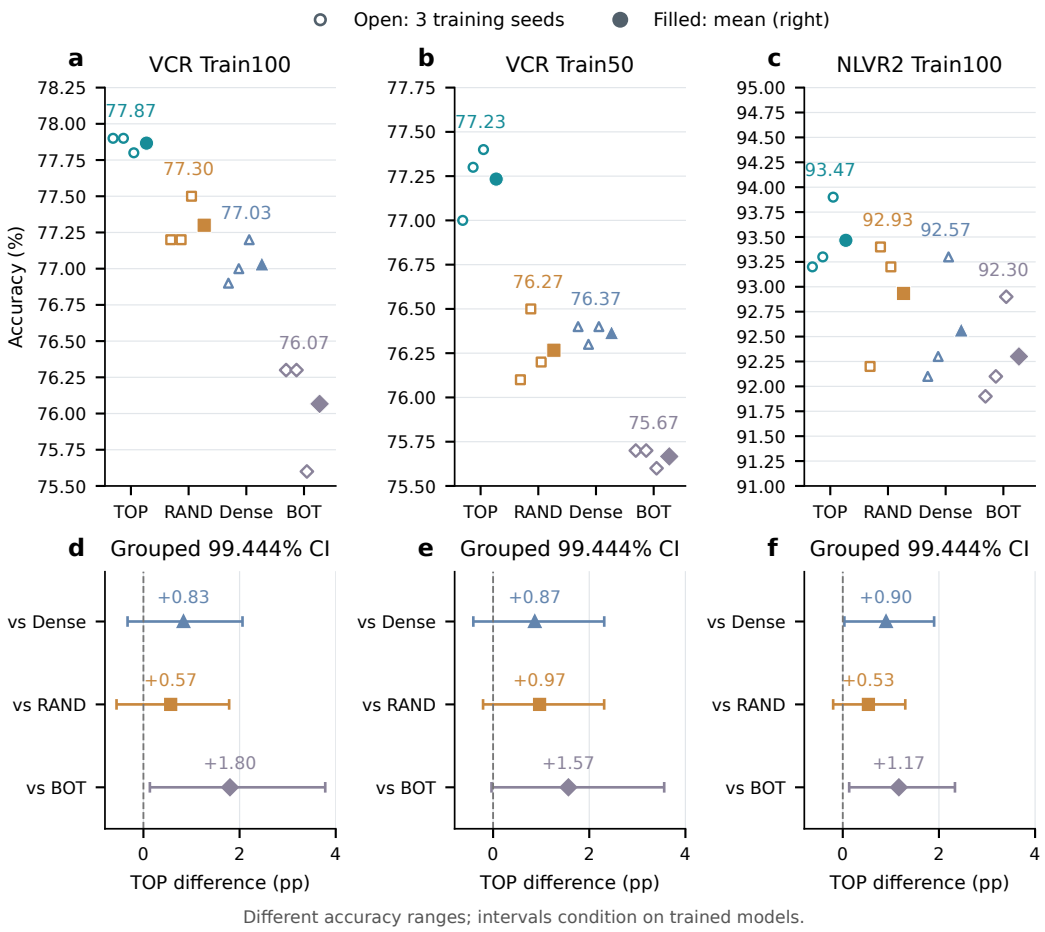

Figure 2: Public-development accuracy and paired uncertainty. Top: small open symbols show three training seeds; filled symbols to their right and numeric labels show means. All three upper panels use separately specified accuracy-axis ranges; horizontal offsets separate seeds and means and have no metric meaning. Bottom: labeled direct TOP-minus-comparator differences with 20,000-replicate intervals. All nine intervals use individual Bonferroni 99.444% levels, retaining VCR movie or NLVR2 statement groups and conditioning on the observed models. The three primary TOP–Dense intervals in the text use a separate 98.333% level. Neither family is a whole-paper correction.

5.3 The mean advantage persists across random allocations, with uncertainty

We expand VCR Train50 from the original RANDOM allocation A to five allocations A–E, each with three training seeds, reusing TOP. Their mean accuracies are 76.267, 76.400, 76.033, 76.400 and 76.100%. The equally weighted mean is 76.240%, versus TOP's 77.233%, yielding +0.993 pp. TOP exceeds each observed allocation mean. The conditional movie-group 95% interval is $[0.219, 1.891]$ pp.

For the fixed collection of three TOP and 15 Uniform-random models, this supports a positive difference under the specified evaluation-group assumptions. It does not integrate over new random expert lists or training outcomes. The historical analysis uses a different sampling procedure, documented in Appendix C.

The observed mean advantage also survives each leave-one-out check. Omitting any one allocation gives +0.942 to +1.033 pp; omitting any one training seed gives +0.910 to +1.050 pp. Thus no single observed list or seed drives the positive mean difference. These ranges describe sensitivity within the tested collection; they are not confidence intervals or additional replications (Appendix C.2).

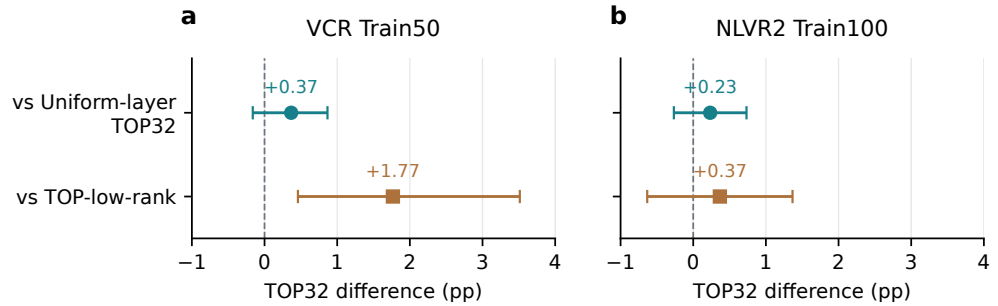

Figure 3: Follow-up quota and rank comparisons. Points are direct TOP32-minus-control differences, averaged across three models per arm. Bars are conditional grouped 98.75% intervals for the four-comparison family; all use a common pp scale. Uniform-layer TOP32 preserves the total adapter budget; TOP-low-rank preserves expert identities but reduces the budget. VCR uses movie groups and NLVR2 statement groups. Historical/new execution compatibility is partially verified, so these comparisons remain provisional.

5.4 Exclusion and positive selection remain unresolved

For Q3, we examine whether avoiding the lowest-mass experts contributes to the difference. NonBottom samples K_l experts from each layer after excluding its frozen BOTTOM set. Let N_{asi} and U_{asi} denote correctness for NonBottom and Uniform-random allocation a, seed s and example i, and T_{si} denote TOP correctness. With equal weighting across five allocations, three seeds and 1,000 examples, and TOP broadcast over allocations,

$$G_{\mathrm{NEG}} = \overline{N - U}, \quad G_{\mathrm{POS}} = \overline{T - N}, \quad G_{\mathrm{TOTAL}} = G_{\mathrm{NEG}} + G_{\mathrm{POS}}. \tag{5}$$

The identity is algebraic. It is not a causal mediation model or a claim that the components are independent.

NonBottom averages 76.527%, between Uniform-random's 76.240% and TOP's 77.233%. Direct estimates are $G_{\mathrm{NEG}} = +0.286667$ and $G_{\mathrm{POS}} = +0.706667$ pp, summing to $+0.993333$ pp. Their conditional movie-group 97.5% intervals are $[0.000, 0.555]$ and $[-0.089, 1.688]$ pp (Figure 4). The first touches zero to numerical precision; we do not interpret it as strict positive exclusion. Neither component establishes an exclusion or positive-ranking mechanism. Historical 97.5% and 98.75% intervals and the original preregistration mismatch are disclosed in Appendix C.

6 Discussion and Limitations

Implications for MoE adaptation. The experiments distinguish evidence about a capacity configuration from evidence about expert ranking. In the tested settings, concentrating higher-rank updates on routing-selected experts yields higher mean accuracy than distributing lower-rank updates across the full expert bank. The repeated-allocation comparison supports a positive conditional difference for the observed model collection, while generalization to new allocations remains uncertain. For evaluating routing-informed MoE adaptation, both controls are informative: all-expert adaptation tests the coverage/rank tradeoff, while repeated random subsets test the routing prior at fixed capacity.

Attribution and alternatives. The follow-up controls favor higher local rank on VCR but do not establish a layer-quota benefit on either task. The TOP7–Dense7 ordering cautions against interpreting expert concentration as uniformly beneficial. Historical execution gaps remain a limit on attribution (Appendix H). ESFT, MoE-Sieve and adaptive-allocation methods have not been compared under a shared protocol, so the results establish no performance ranking against them. NRG contributes an empirical analysis of inherited routing-selection ideas under low-rank allocation.

Scope and statistical uncertainty. Evaluation covers one MoE backbone on low-data multimodal development subsets. Section 3.4 establishes budget relations for larger compatible expert banks,

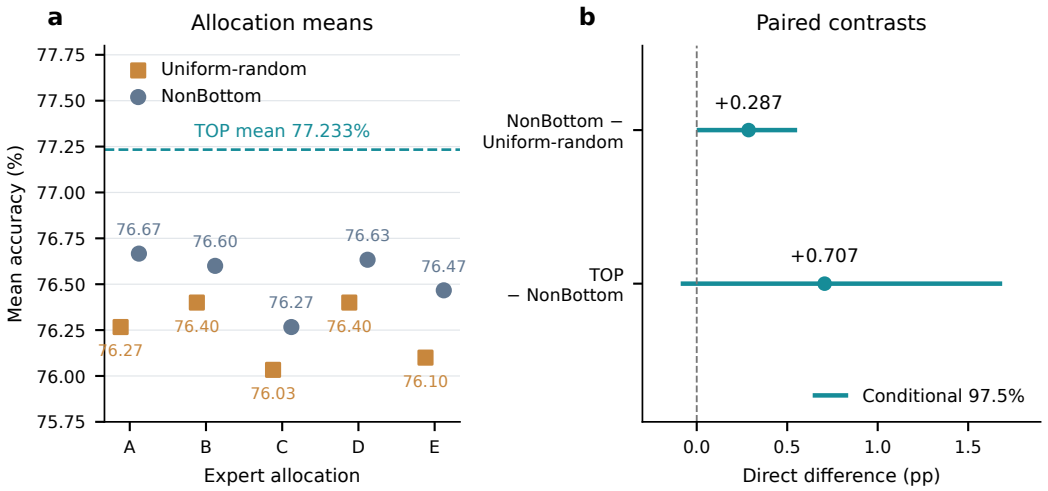

Figure 4: Allocation sensitivity and an inconclusive NonBottom probe on VCR Train50. Left: each point averages three training seeds; A–E are different expert allocations, not a time series. The line is the reused TOP mean. Right: observed decomposition components, with conditional movie-group 97.5% intervals for the two-component family. The first component touches zero; the second crosses it. Historical intervals and the different resampling procedures are compared in Appendix C. Neither the positive means nor their algebraic sum establish a mechanism.

while accuracy on larger or different MoE backbones, language-only tasks and independent test sets remains untested. Three training seeds, five random allocations and repeated development-set use limit generalization. The reanalysis preserves chosen evaluation groups but conditions on observed models. Three seeds and five lists do not establish seed- or allocation-population coverage; group exchangeability and residual dependence remain limitations. Historical product-index sampling differs from the preregistered hierarchical design. Appendix C records these assumptions and interval levels. There is no whole-paper multiplicity correction. Threshold, rank and calibration choices define the tested protocol rather than a demonstrated optimum, and VCR Train50 versus Train100 changes both data and update count.

Historical evidence and data provenance. Function A supplies supporting observations of rank sensitivity, including a rank-one failure and favorable TOP ordering at higher ranks. These are single-seed historical runs with runtime changes, which prevent attributing the pattern to rank alone. Current Hard72 includes a clip exposed in historical Train100 and routing calibration, and label-generation provenance remains incomplete. Appendix D preserves these qualifications; the public-task comparisons carry the main empirical argument.

Parameter budgets and system costs. The budget measures trainable adapter parameters. It does not establish training speed, activation memory or end-to-end cost: routing calibration, communication, offloading and the sparse-update implementation also contribute. Systematic cost measurements are outside the present evidence.

7 Conclusion

The tested routing-selected rank-32 configuration has higher means than approximately budget-matched all-expert adaptation across three settings. Yet selected low-rank adaptation falls below all-expert adaptation on VCR, and neither task establishes a layer-quota benefit. These findings motivate separate tests of capacity distribution and expert identity. Historical execution gaps, fixed-model inference and one-backbone development evaluation limit generalization; the budget derivation extends accounting, not empirical accuracy claims.

AI USE STATEMENT

Generative AI tools assisted with hypothesis refinement, experimental-control design, mathematical budget exposition, manuscript preparation, literature discovery, interpretation of recorded results, and organization and verification of scoring and statistical reproduction materials. The manuscript-preparation workflow used preserved experimental predictions and did not create synthetic benchmark observations. Verification included prediction identity and scoring, reproduction of direct estimates and grouped intervals, and source and artifact hashes. These checks do not replace scientific review by the author. The author remains responsible for the final scientific content and the accuracy of AI-assisted text, claims and artifacts.

REPRODUCIBILITY STATEMENT

Section 3 defines the allocation protocol and parameter budgets; Section 4 specifies the evaluation objects and statistical assumptions. The appendices provide training settings, seed-level results, exact resampling conventions and provenance limitations. A local evidence archive preserves configurations, selection manifests, scoring/statistical implementations and aligned predictions. Trained adapters, training shards and merged weights are no longer retained; checkpoint-based inference reproduction is therefore unavailable. An anonymous numerical reproduction bundle accompanies this paper. It reparses the original 63,000 saved outputs and the 24,000 outputs used by the follow-up comparisons, which reuse 12 original models: 75 unique models and 75,000 unique model–example outputs in total. It aligns evaluation identities, reconstructs inferential inputs and recomputes the statistics. Source and projection hashes document provenance; no images or prompts are included. This reproduces scoring and statistical calculations, not full training or independent validation of ground-truth labels. The bundle removes private infrastructure identifiers and supplies release-specific hashes. Original dataset rights remain with their respective owners.

ETHICS AND DATA PROVENANCE

The public-task results use VCR and NLVR2 development data under their original distribution conditions. Function A is an internal simulation task; its full label-generation provenance and redistribution conditions are not yet established. Its historical overlap and identity limitations are disclosed in Appendix D. The present results are not a deployment-safety evaluation. No assumption is made about human-review proportions or unseen data licensing.

REFERENCES

Guanzhi Deng, Bo Li, Ronghao Chen, Xiujin Liu, Zhuo Han, Huacan Wang, Lijie Wen, and Linqi Song. DR-LoRA: Dynamic Rank LoRA for Fine-Tuning Mixture-of-Experts Models. arXiv:2601.04823v6, 2026. URL https://arxiv.org/abs/2601.04823v6. Version v6.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of ACL, Volume 1, 1383–1392*, 2018. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128/.

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.

Haoze He, Xingyuan Ding, Xuan Jiang, Xinkai Zou, Alex Cheng, Yibo Zhao, Juncheng Billy Li, and Heather Miller. Preserving Long-Tailed Expert Information in Mixture-of-Experts Tuning. arXiv:2604.23036v2, 2026. URL https://arxiv.org/abs/2604.23036v2. Version v2.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685v2, 2021. URL https://arxiv.org/abs/2106.09685v2. Version v2.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-Decomposed Low-Rank Adaptation. In *Proceedings of ICML, PMLR 235, 32100–32121*, 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

Andrea Manzoni. MoE-Sieve: Routing-Guided LoRA for Efficient MoE Fine-Tuning. arXiv:2603.24044v1, 2026. URL https://arxiv.org/abs/2603.24044v1. Version v1.

Qwen Team. Qwen3.5-35B-A3B-Base model card. Hugging Face model repository; accessed 11 September 2026, 2026. URL https://huggingface.co/Qwen/Qwen3.5-35B-A3B-Base.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of ACL, 6418–6428*, 2019. doi: 10.18653/v1/P19-1644. URL https://aclanthology.org/P19-1644/.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. Let the Expert Stick to His Last: Expert-Specialized Fine-Tuning for Sparse Architectural Large Language Models. In *Proceedings of EMNLP, 784–801*, 2024. doi: 10.18653/v1/2024.emnlp-main.46. URL https://aclanthology.org/2024.emnlp-main.46/.

Qingyu Yang, Haonan He, Minglei Li, Jingqi Ye, Tao Chen, Lei Bai, and Peng Ye. MoE2-LoRA: When MoE Models Meet MoE-style Low-Rank Adaptation. arXiv:2607.21978v1, 2026. URL https://arxiv.org/abs/2607.21978v1. Version v1.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *Proceedings of CVPR, 6720–6731*, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zellers_From_Recognition_to_Cognition_Visual_Commonsense_Reasoning_CVPR_2019_paper.html.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

A Contribution comparison and untested alternatives

Table 4: Relationship to prior work. This is a design comparison, not a performance ranking. NRG has not been run against these implementations under a unified protocol.

Work	Relevant design	Relationship to NRG
LoRA (Hu et al., 2021)	Low-rank weight updates	Inherited parameterization
DoRA (Liu et al., 2024)	Magnitude/direction decomposition	Inherited parameterization; not the allocation contribution
AdaLoRA (Zhang et al., 2023)	Importance-based budget adaptation across matrices	Related allocation problem; NRG fixes membership before SFT
ESFT (Wang et al., 2024)	Native scores, cumulative layer selection, same-count random control	Direct intellectual predecessor; these ideas are not new here
MoE-Sieve (Manzoni, 2026)	Routing-count profiling and static expert-subset LoRA	Close prior art; NRG differs in the investigated protocol, not the basic combination
DR-LoRA (Deng et al., 2026)	Routing/gradient-informed rank growth	Dynamic capacity alternative not evaluated here
MoE2-LoRA (Yang et al., 2026)	Routing-conditioned dynamic adapters and a shared pool	Broader routing/adapter design; no direct performance comparison

The additional layer-quota and fixed-membership rank controls analyze these inherited choices under a low-rank budget; they do not introduce a new routing or adaptation module. Literature metadata were checked against primary publication pages or versioned author preprints. The DR-LoRA record used here is the August 2026 revision, whose title and author list differ from its initial version. The arXiv version of MoE2-LoRA is a preprint, not an assumed accepted conference paper.

B Detailed training and evaluation record

Table 5: Original-cohort implementation settings; follow-up arms are detailed in Appendix H. Identical configuration fields do not imply equal wall-clock cost or a fully identical distributed execution history.

Item	VCR	NLVR2
Backbone	Qwen3.5-35B-A3B-Base	Same
Epochs / learning rate	3 / 5e-5	3 / 5e-5
Scheduler / warmup	Cosine / 0.03	Same
Weight decay / max grad norm	0 / 1.0	Same
Per-device batch / accumulation	1 / 1	Same
Precision / gradient checkpointing	bfloat16 / enabled	Same
Sequence cutoff	3,072	3,072
Image pixel bounds	1,024–589,824	1,024–589,824 per processor configuration
Sparse rank / alpha	32 / 64	32 / 64
Dense rank / alpha	7 / 14	8 / 16
Training seeds	42; 1066738582; 2037739350	42; 1958630825; 2599520754
Decoding temperature / top-p / top-k	0 / 1 / 1	0 / 1 / 1
Maximum answer tokens	16	8
Decoding seed	20260907	42
Images per input	1	2

The five VCR allocation seeds recorded for the robustness/probe series are A=20260907, B=3799993942, C=2791634881, D=2165811378 and E=2103443103. Training seeds and allocation seeds index different sources of variation. P7 NonBottom lists were checked to be disjoint from

the frozen BOTTOM list in each layer and to retain the same K_l. Full lists are supplied in the evidence package; a list regenerated with another RNG implementation is a different experimental allocation.

Table 6: Individual seed accuracies (%). Seed1/2/3 map to the task-specific seed lists in the training-settings table in this appendix.

Regime	Arm	Seed1	Seed2	Seed3
VCR Train100	TOP	77.9	77.9	77.8
VCR Train100	RANDOM	77.2	77.2	77.5
VCR Train100	DENSE	76.9	77.0	77.2
VCR Train100	BOTTOM	76.3	76.3	75.6
VCR Train50	TOP	77.0	77.3	77.4
VCR Train50	RANDOM	76.1	76.5	76.2
VCR Train50	DENSE	76.4	76.3	76.4
VCR Train50	BOTTOM	75.7	75.7	75.6
NLVR2 Train100	TOP	93.2	93.3	93.9
NLVR2 Train100	RANDOM	92.2	93.4	93.2
NLVR2 Train100	DENSE	92.1	92.3	93.3
NLVR2 Train100	BOTTOM	91.9	92.1	92.9

Execution audit. All 63 unique public-task runs have acquired configurations, training logs and trainer states: 24 Train100 runs record 39 updates, and 39 Train50 runs record 21 updates. Each records three completed epochs, eight processes, global batch eight, and per-device batch/accumulation one. The update counts are consistent with $3\lceil N/8 \rceil$. This arithmetic does not prove which examples were padded or their order. Recorded learning-rate histories agree within each training size across arms and seeds. Summed post-rebinding trainable counts equal the budgets in Table 1 in every run. No per-example/per-epoch exposure trace survives, and the full historical weight/environment state was not reconstructed.

Runtime parameter mapping. The scoped bridge overrides template projection targets with the packed routed-expert gate-up and down parameters. For each expert, fused gate-up has shape $[2d, h]$ and down has shape $[h, d]$; each uses factors $A \in \mathbb{R}^{r \times \text{input}}$, $B \in \mathbb{R}^{\text{output} \times r}$ and output-channel magnitudes. The gate and up halves share the fused input-side factor. The bridge implements magnitude scaling of the updated weight using row norms computed in float32. After AutoEP replaces parameter objects, it rebinds the optimizer to the selected local parameters; the audit checks all eight rank-specific rebinding records per run. Configuration intent, runtime source and logged optimizer totals agree. These checks do not substitute for replaying the complete distributed training computation.

Input identity. Metadata-only remote checks confirm VCR Train50 is a subset of Train100, with no shared movie identifiers or image basenames between these training subsets and the frozen development set. VCR calibration uses the Train100 input manifest with reference labels removed. NLVR2 training and development image basenames have no intersection. These are identifier-level checks, not pixel-duplicate or pretraining-contamination audits. Full training/dev sample contents were not transferred for this execution audit.

C Exact statistics and deviations from the planned analysis

The secondary rows are outside the four-comparison follow-up family. No combined six-comparison familywise coverage is claimed. The original and follow-up analyses use different documented endpoint conventions; the table does not redefine frozen statistics.

C.1 Conditional reanalysis and sensitivity

The analysis plan was fixed before computing the new intervals, after reviewing historical results. It is therefore a post-hoc sensitivity analysis. For each evaluation item i, let d_i be the paired correctness

Table 7: Index of current conditional intervals. All rows hold the observed models and expert lists fixed. V=VCR movie groups; N=NLVR2 statement groups. N image-set and shared-item analyses are sensitivity checks, not independent replications.

Comparison	Family m	Individual CI	Groups	Location
TOP–Dense, three settings	3	98.333%	V / N	Sec. 5.1; Table 8
TOP vs Dense/RANDOM/BOTTOM	9	99.444%	V / N	Fig. 2
NonBottom decomposition components	2	97.5%	V	Fig. 4
TOP–Uniform-random total	Descriptive	95%	V	Sec. 5.3
Follow-up quota/rank contrasts	4	98.75%	V / N	Fig. 3
Follow-up same-rank TOP–Dense	Secondary	98.75%	V / N	Table 14

difference averaged over the observed trained models. For group g, let $D_g = \sum_{i \in g} d_i$ and $n_g = |g|$. Drawing G groups with replacement yields multiplicities w_g and

$$\widehat{\Delta}^* = 100 \frac{\sum_g w_g D_g}{\sum_g w_g n_g}. \tag{6}$$

This ratio retains example weighting for unequal movie groups. It is not an equally weighted mean of movie accuracies. Common multiplicities apply to every arm and decomposition component. VCR has 181 movie groups per subset; NLVR2 has 250 statement groups of four examples, nested within 226 image sets. Statement grouping retains its four examples; image-set grouping is a coarser sensitivity check. Neither proves all residual dependence absent.

For P7, first average TOP across its three models and Uniform-random/NonBottom across their respective 15 models. TOP is not treated as 15 independent models. Common resampling weights preserve the decomposition identity in every replicate. A shared-item reference draws one set of item indices for all observed models. Neither procedure resamples a population of training seeds or expert allocations. The bundle reports per-seed, per-allocation and leave-one-out summaries descriptively.

For the original-cohort conditional reanalysis, the reproducible implementation uses NumPy's generator, 20,000 draws and interpolated percentile endpoints. Its base seed is 2026091301, incremented in the analysis order recorded in the locked plan. The bundle stores each analysis seed, all four requested interval levels and input hashes. Bonferroni families are three TOP–Dense comparisons (98.333%), all nine plotted TOP comparisons (99.444%), and two P7 components (97.5%). The total contrast uses descriptive 95% intervals. No whole-paper correction or prospective confirmatory status is claimed.

Table 8: Conditional TOP–Dense sensitivity, in pp. All intervals use 98.333%, for the three-setting family. Models and expert lists are held fixed.

Setting	Resampling unit	Estimate	Lower	Upper
VCR Train100	item	0.833	-0.033	1.733
VCR Train100	movie	0.833	-0.150	1.885
VCR Train50	item	0.867	-0.233	2.000
VCR Train50	movie	0.867	-0.239	2.083
NLVR2 Train100	item	0.900	0.133	1.733
NLVR2 Train100	statement	0.900	0.133	1.733
NLVR2 Train100	image set	0.900	0.136	1.762

C.2 Leave-one-out sensitivity

For each omission, we recompute the direct difference with equal weighting over the remaining observed lists or seeds. Omitting a list removes its Uniform-random and NonBottom models but leaves the fixed TOP models unchanged. Omitting a seed removes that seed from every method. The ranges summarize the finite set of omissions; they are not percentile intervals or estimates of population coverage.

Table 9: VCR Train50 leave-one-out differences in pp. All direct differences are recomputed from the same frozen correctness arrays.

Contrast	Full estimate	Omit one allocation	Omit one seed
G_NEG	0.287	[0.258, 0.308]	[0.230, 0.320]
G_POS	0.707	[0.642, 0.742]	[0.680, 0.740]
G_TOTAL	0.993	[0.942, 1.033]	[0.910, 1.050]

C.3 Historical procedures and deviations

For public four-arm comparisons, the observed contrast is the mean of the paired correctness difference across the three seeds and 1,000 items. P5C/P5D bootstrap resample three seed labels with replacement and then resample 1,000 indices within each chosen seed. They use 20,000 replicates with the Python random generator seeded at 20260908. The original program consumes the RNG consecutively across comparisons. Its nominal 95% endpoints are sorted-replicate positions 500 and 19499 using zero-based indexing, rather than NumPy's default interpolated quantiles. Changing the RNG sequence, comparison order or quantile convention need not reproduce the frozen endpoints.

P8 uses NumPy's generator with seed 103184856 and 20,000 replicates. For every replicate it first draws three training-seed indices, then draws 1,000 sample indices for each selected seed, using common indices across arms. Nominal endpoints are quantiles 0.025 and 0.975. The three-TOP-comparison Bonferroni endpoints are 0.0083333333 and 0.9916666667, corresponding to an individual interval level of 98.333333%. Although the original output also contains other pairwise intervals, an m=3 adjustment cannot establish familywise coverage for all six pairs. Group consistency is the fraction of 250 groups in which all four examples are correct; it remains descriptive and was not used as a cluster-bootstrap substitute.

P6-R1 uses a direct mean over RANDOM allocations A–E, with TOP reused over the allocation axis. The primary direct difference is +0.993333 pp and the stored 95% interval is [-0.100,2.100] pp. Its NumPy generator seed is 157454169. Each of 20,000 replicates samples 1,000 independent triples of allocation, training seed and sample. It pairs the two methods within a triple but does not draw and retain full training-model clusters. This implementation cannot be described as a full hierarchical bootstrap merely because all three axes are sampled. The manuscript preserves its interval with that limitation and does not use the superseded B–E-only result.

For P7, direct sums of the original binary arrays give G_NEG=43/15000 and G_POS=53/7500 in accuracy units; their sum is G_TOTAL=149/15000. The bootstrap seed is 2491244670, with 20,000 replicates and the same product-index structure. The output means of +0.289450, +0.700245 and +0.989695 pp are means of bootstrap draws. We replace them in point-estimate reporting with the observed +0.286667, +0.706667 and +0.993333 pp, without changing predictions or stored intervals.

P7 preregistration names two co-primary contrasts, familywise alpha=0.05 and a 97.5% two-sided hierarchical percentile interval. A per-comparison 97.5% interval already supplies the Bonferroni level for two comparisons at this alpha, assuming the underlying intervals are otherwise valid. The frozen script additionally computes endpoints 0.00625 and 0.99375 and labels them adjusted, yielding an actual 98.75% level. This is both a reporting discrepancy and, together with the product-index sampling, an implementation deviation. The 97.5% field matches the requested quantile level, but not the requested hierarchical sampling structure. Reporting it does not restore full preregistration conformity.

Table 10: P7 point estimates and stored intervals, in pp. G_TOTAL is descriptive, not a third co-primary comparison.

Contrast	Direct estimate	Bootstrap mean	Actual 97.5% CI	Actual 98.75% CI
G_NEG	+0.286667	+0.289450	[-0.600, 1.200]	[-0.700, 1.300]
G_POS	+0.706667	+0.700245	[-0.500, 1.900]	[-0.600, 2.000]
G_TOTAL	+0.993333	+0.989695	[-0.20125, 2.300]	Not defined

No whole-paper multiplicity correction is applied. The probability that a bootstrap draw is positive is not reported as a frequentist p-value. Overlapping datasets and reused TOP/Uniform-random predictions mean that P5D, P6-R1 and P7 are related analyses, not three independent replications. All interval exclusions in the text refer to their specified procedures; none establishes absence of cluster effects, data-selection effects or model-pretraining contamination.

During local evidence preparation, P5C/P5D correctness cells, seed accuracies and direct contrasts were checked against the alignments; their bootstrap code was inspected but not rerun. P6-R1, P7 and P8 frozen bootstrap intervals were reproduced locally with NumPy 2.4.3. P8's 12 prediction files were hash-checked, aligned on identifiers/group labels/targets, and their group consistency counts reproduced. Those earlier checks reproduced historical calculations. The conditional reanalysis above is a new statistical analysis of the same predictions, not a new model evaluation.

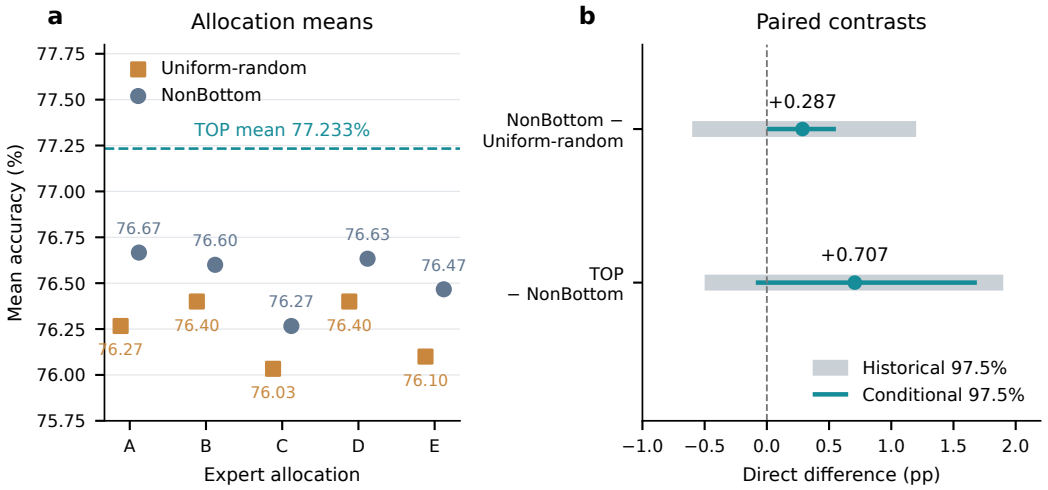

Figure 5: Historical versus conditional P7 intervals. Right: both displayed intervals have individual 97.5% levels, but the historical product-index procedure (gray) and the current fixed-model movie-group procedure (teal) have different sampling targets. Their widths cannot be interpreted as an improvement in inference about the same population. Left: observed allocation means, repeated here for context.

D FUNCTION A DATA PROVENANCE AND HISTORICAL INTERPRETATION

The source handoff identifies Function A as internally generated simulation data and names box_pos.json and pile_pos.json as visual-state ground-truth inputs. The local manifests support the simulation directory provenance, but the complete label-generation program, simulator version and human-review proportions were not independently reconstructed. In particular, a correct visual-state source does not automatically certify the full ordered task_queue labeling process. We therefore do not describe these labels as independently human-validated ground truth or the dataset as publicly released real-world observations.

The four acquired manifests contain respectively 100, 500, 200 and 72 unique clip IDs and use front120/back120 image paths. The current Dev200 content has a recorded SHA256, but the reason for its disagreement with an earlier historical identity remains unresolved. Current path existence cannot certify the exact bytes read by every historical run.

We preserve the original Hard72 identity and do not silently remove clip_3047 or rescore the set as if 71 examples were the preregistered evaluation. Later Train500 stress results are supporting development evidence, not an independent external holdout claim. The rank plot similarly preserves the historical negative rank-one observation without treating later runtime changes as a clean randomized intervention on rank alone.

Table 11: Current clip-ID overlap audit. This audit parses the clip identifier from image paths, not the image pixels; it does not rule out near-duplicate scenes under different clip IDs.

Current split pair	Shared clip IDs	Interpretation
Train100 / Dev200	0	No current clip-ID intersection
Train500 / Dev200	0	No current clip-ID intersection
Dev200 / Hard72	0	No current clip-ID intersection
Train500 / Hard72	0	No current Train500 clip overlap
Train100 / Hard72	1: clip_3047	Historical supervised exposure
Calibration profile / Hard72	1: clip_3047	Calibration exposure remains relevant
Train100 / Train500	99	Not a strict inclusion of all Train100 examples

E PREDICTION-TO-STATISTICS REPRODUCTION

The bundle separates frozen evidence, reproducible calculations and interpretation. A byte hash establishes identity, not semantic correctness. The scoring entry point reads minimal projections of the 63 original public-task prediction files, reparses raw output strings using the recorded VCR/NLVR2 contracts, and checks the saved prediction, validity and correctness fields. Invalid answers count as wrong. It rejects duplicate, missing or extra identifiers and inconsistent labels or groups, then aligns records to the frozen evaluation order.

Table 12: Reproduction map. Paths are relative to the portable bundle. The local provenance index links projections to original files by SHA256.

Claim or check	Input, entry point and output
Answer scoring	Raw-output records in `data/predictions/`; `implementation/score_predictions.py`; `scoring_results/run_scores.csv`
Identity and grouping	`data/evaluation_keys/` and `data/scoring_manifest.json`; scorer rejects identity, label and group mismatches
Array reconstruction	Scorer writes `scoring_results/correctness.npz` and checks exact array equality with the frozen inferential inputs
Intervals and leave-one-out	`implementation/reproduce.py` accepts rebuilt arrays; outputs interval and diagnostic files under the chosen result directory
Allocation and execution	`data/selection_lists.json`, `data/execution_protocol.csv` and `data/execution_evidence.json` retain identities, budgets and recorded schedules

All 63,000 outputs were rescored, with zero invalid outputs and exact equality of the 12 rebuilt correctness/grouping arrays. Recomputed intervals and descriptive diagnostics match the frozen reanalysis results. The historical unadapted-base array is not part of these inferential analyses and is not rebuilt by this entry point. Gold labels and grouping keys come from frozen alignments; this verifies scoring consistency, not independent relabeling, image content or pretraining contamination. The original-cohort subpackage contains no images, prompts or model weights and does not recreate historical training. Full raw sources and acquisition hashes remain in the local evidence archive; the distributed metadata are anonymized projections, while original acquisition hashes remain in the local archive.

The paper uses the original task-specific allocation and closure records; acquisition and correction history remain in the local provenance archive. The trained adapters, training shards and merged weights were subsequently removed during experiment closure. The retained prediction and metadata archive supports scoring/statistical reproduction, but it cannot support reconstruction of those trained checkpoints. P7's original sampling and interval-label deviations remain in Appendix C, and Function A's provenance limitations remain in Appendix D. No Test2416 data were accessed during manuscript preparation.

F General budget derivation

For a fused gate-up matrix of shape $[2d_l, h_l]$, the factors contain $r_l(h_l + 2d_l)$ parameters and its output magnitudes contain $2d_l$. The down matrix adds $r_l(h_l + d_l)$ factor parameters and h_l magnitudes, giving Equation 4.

For homogeneous layers, define $\rho = \sum_l K_l/(LE)$. Exactly matching an all-expert rank-r_d budget gives the continuous sparse rank

$$r_s^* = \frac{r_d}{\rho} + \frac{m}{c}\left(\frac{1}{\rho} - 1\right). \tag{7}$$

Implementation requires a feasible integer rank and explicit rounding mismatch. Equation 7 shows how reduced coverage can support greater local rank without depending on L or E separately.

At fixed expert widths, rank and selected fraction, adding layers or experts increases both adapter and routed-expert weight counts linearly. Scaling both widths by s increases adapter count linearly in s, while the corresponding bias-free expert matrices, containing $3h_l d_l$ weights per expert, grow quadratically. This reduces the adapter-to-expert-weight ratio, not necessarily absolute adapter cost. Larger expert inventories also need not yield a smaller selected fraction: a uniform profile gives $K_l = \lceil \tau E_l \rceil$. Each new backbone therefore requires its own routing profile and budget calibration. These relations establish architectural applicability to larger compatible MoEs, conditional on the stated parameterization; they predict neither accuracy gains nor runtime or memory savings.

G Layer allocations and historical rank observations

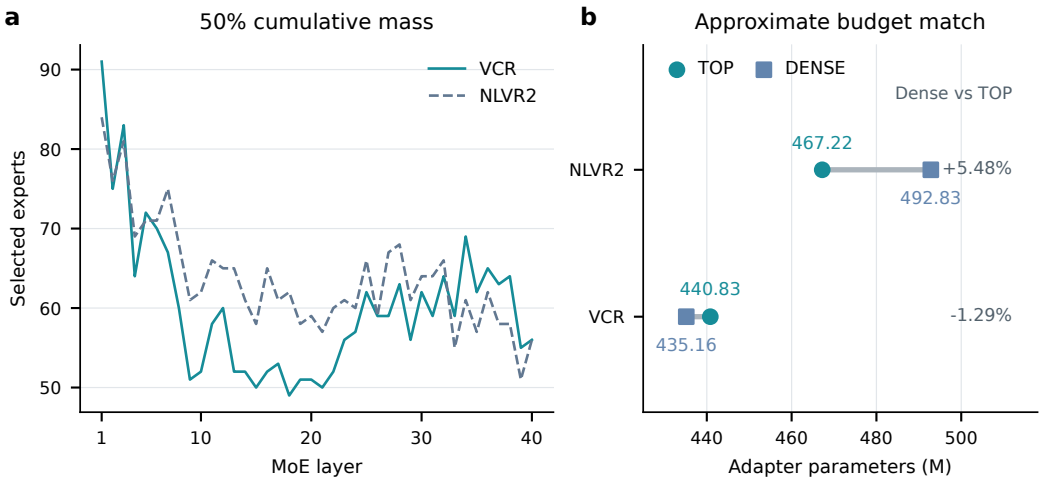

Figure 6: Frozen per-layer counts and adapter budgets. Counts follow cumulative routing mass at threshold 0.5; budgets retain the integer-rank mismatch. Connected endpoints compare TOP and Dense parameter counts within each task; percentages give $(P_{\text{Dense}} - P_{\text{TOP}})/P_{\text{TOP}}$. Neither panel measures memory or training speed.

The rank-one TOP/RANDOM/BOTTOM values are 14/12/15%. Later rank8, rank16 and rank32 values are 62/30.5/13.5%, 88/54/23%, and 97.5/76.5/27.5%, respectively. CORE checks parsing/schema validity, exact uncertainty information, and the exact ordered task queue; it does not require exact matching of free-text reasoning. These historical observations have lower inferential strength than the public-task seed replications.

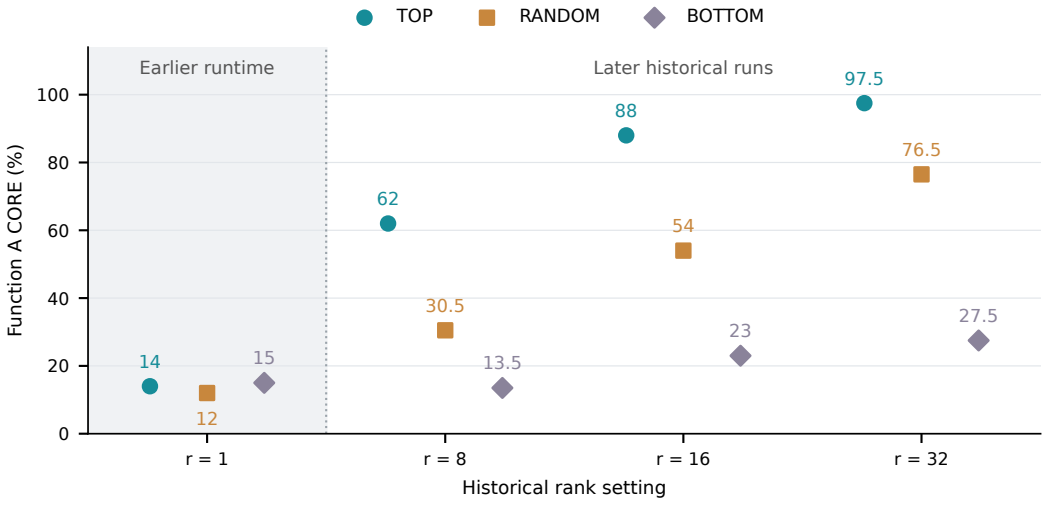

Figure 7: Historical Function A rank settings: single-seed CORE results on 200 examples. Unconnected points represent discrete historical observations; the shaded region separates the earlier runtime. Horizontal offsets distinguish methods and have no metric meaning. The initial rank-one experiment precedes subsequent runtime changes. Rank32 is the best tested setting under the recorded criterion, not a globally optimal rank.

H FOLLOW-UP LAYER-QUOTA AND LOCAL-RANK CONTROLS

H.1 DESIGN AND PARAMETER ACCOUNTING

The follow-up study adds two arms to VCR Train50 and NLVR2 Train100, each with three training seeds. It reuses frozen task-specific profiles without new profiling. Let $S = \sum_l K_l$ be the original number of selected layer–expert slots. With zero-based layer index $l = 0, \ldots, 39$, Uniform-layer TOP uses

$$K_l^{\text{uniform}} = \left\lfloor \frac{(l+1)S}{40} \right\rfloor - \left\lfloor \frac{lS}{40} \right\rfloor . \tag{8}$$

This gives 35 layers with 60 and five with 61 slots for VCR ($S = 2405$), and 11 layers with 63 and 29 with 64 for NLVR2 ($S = 2549$). Within-layer experts are selected by descending frozen routing score with deterministic expert-ID tie-breaking. It tests the quota rule as a whole, including the membership changes induced by reallocating slots; it does not hold expert identity fixed across layers. This control is distinct from Uniform-random expert selection at the original layer counts.

TOP-low-rank reuses exactly the original TOP expert identities and layer counts. Rank and alpha are 7/14 for VCR and 8/16 for NLVR2, versus 32/64 for TOP32. Constant α/r does not make the two parameterizations or optimization paths identical. The lower-rank comparison changes both rank and total trainable parameters.

Table 13: Follow-up accuracy (%) and budgets. Each mean and sample SD uses three fixed trained models. Seed columns follow the task-specific order below. Historical TOP32/Dense predictions are reused, not new runs.

Task	Arm	Params (M)	Seed1	Seed2	Seed3	Mean ± SD
VCR50	TOP32 (historical)	440.827	77.00	77.30	77.40	77.23 ± 0.21
VCR50	Uniform-layer TOP32	440.827	76.90	76.60	77.10	76.87 ± 0.25
VCR50	TOP-low-rank	102.203	75.50	75.70	75.20	75.47 ± 0.25
VCR50	Dense (historical)	435.159	76.40	76.30	76.40	76.37 ± 0.06
NLVR2	TOP32 (historical)	467.222	93.20	93.30	93.90	93.47 ± 0.38
NLVR2	Uniform-layer TOP32	467.222	92.40	93.60	93.70	93.23 ± 0.72
NLVR2	TOP-low-rank	122.678	93.20	93.40	92.70	93.10 ± 0.36
NLVR2	Dense (historical)	492.831	92.10	92.30	93.30	92.57 ± 0.64

VCR seeds are 42, 1066738582 and 2037739350; NLVR2 seeds are 42, 1958630825 and 2599520754. Historical NLVR2 uses the original seed1 plus both subsequent seed2/seed3 runs. Recorded configs, selection audits and runtime manifests are included in the supplementary evidence package. The stated recipe follows Section 4; this does not certify identical individual sample exposure or dynamically identical optimizer behavior across historical and new runs.

H.2 STATISTICAL CONTRACT AND COMPLETE CONTRASTS

Each comparison aligns all 1,000 IDs and checks gold labels and group keys across all three models in every arm. Raw strings are reparsed independently of the stored normalized answer; no invalid outputs occur. Paired correctness is averaged over the three fixed models before resampling whole groups with common weights across methods. VCR has 181 movie groups; NLVR2 has 250 statement groups and a 226-image-set sensitivity analysis. Models and training seeds are not resampled. This estimates example-weighted differences for the observed model collection under exchangeable-group assumptions.

The four primary contrasts are TOP32 minus Uniform-layer TOP32 and TOP32 minus TOP-low-rank in each task. Their individual 98.75% levels implement a Bonferroni adjustment for four comparisons, not for all analyses in the paper. The frozen implementation uses Python's `random.Random(20260914)` independently reset for each contrast, 20,000 draws, sorted group labels and zero-based sorted-draw indices 124 and 19874. Resetting the same generator preserves common group draws across contrasts within a task. NLVR2 image-set sensitivity uses seed 20260915. These order-statistic endpoints differ from interpolated NumPy quantiles used in the ear-

lier conditional reanalysis. The portable scorer reproduces the frozen convention without changing its endpoints.

Table 14: All follow-up contrasts in pp. Primary intervals belong to the four-comparison family. Secondary contrasts retain individual 98.75% intervals, but are outside that family; no six-comparison familywise claim is made.

Task	Contrast	Estimate	98.75% interval
VCR50	TOP32 − Uniform-layer TOP32	$+0.367$	$[-0.162, 0.866]$
VCR50	TOP32 − TOP-low-rank	$+1.767$	$[0.459, 3.513]$
VCR50	TOP-low-rank − Dense (secondary)	-0.900	$[-2.001, -0.072]$
NLVR2	TOP32 − Uniform-layer TOP32	$+0.233$	$[-0.267, 0.733]$
NLVR2	TOP32 − TOP-low-rank	$+0.367$	$[-0.633, 1.367]$
NLVR2	TOP-low-rank − Dense (secondary)	$+0.533$	$[-0.533, 1.633]$

NLVR2 image-set intervals are $[-0.264, 0.730]$ pp for TOP32 minus Uniform-layer TOP32 and $[-0.551, 1.302]$ for TOP32 minus TOP8. Neither excludes zero. Omitting one matched seed gives VCR ranges $[0.200, 0.500]$ pp (quota contrast) and $[1.550, 1.900]$ (rank contrast); NLVR2 ranges are $[-0.050, 0.500]$ and $[-0.050, 0.600]$. These ranges describe three omissions, not confidence intervals or seed-population inference.

H.3 Historical compatibility and reproduction limits

Evaluation ID, gold, grouping and scoring checks pass; model configuration and index identities agree. Those checks do not establish base-weight byte identity or complete processor/runtime equivalence. All 12 new merge barriers and reopen checks passed at execution time. However, historical/new dynamic DoRA initialization and scaling, optimizer parameter rebinding, complete distributed configuration equivalence and actual per-sample training exposure are not fully evidenced. Some NLVR2 decoder fields were not repeated in the new manifests. The archived compatibility status therefore remains provisional with explicit unknowns. It is neither evidence that training was unfair nor proof that the compared execution paths were identical.

The entry point `implementation/reproduce_supplement.py` checks all 24 original prediction-file hashes, independently reparses raw strings, verifies every model's IDs/gold/groups, and reproduces direct means, grouped intervals and leave-one-seed-out values. Twelve of these models overlap the original 63-model package; the supplement adds 12 unique models. It preserves the final comparison JSON and both design audits alongside archived implementation/configuration records. The retained local records cannot replace deleted trained weights or recover unrecorded dynamic execution traces. Thus the new controls narrow the empirical interpretation without closing all implementation-level attribution questions.

I ALLOCATION PSEUDOCODE AND REPRODUCIBILITY SCOPE

Inputs and outputs. Inputs are the reference-answer-free routing profile q_{le}, threshold $\tau = 0.5$, expert matrix widths, rank $r_s = 32$ with $\alpha_s = 64$, and frozen expert lists where available. Outputs are layer quotas, expert identities, ranks, alpha values and exact adapter parameter counts. The following is a specification of the allocation stage, not a reconstruction of the historical distributed runtime.

1. For each layer, rank experts by descending q_{le} and take the smallest prefix whose cumulative mass reaches τ. Record its length K_l and expert IDs. The frozen TOP manifest is authoritative for historical membership.
2. At the same K_l and rank 32, use the highest-mass IDs for TOP, lowest-mass IDs for BOTTOM, and uniformly sampled distinct IDs for RANDOM. NonBottom samples distinct IDs outside the frozen BOTTOM set. Reuse recorded lists across training seeds; a fresh random draw is a new allocation.
3. Compute $S = \sum_l K_l$ and $P_s = \sum_l K_l(c_l r_s + m_l)$ with the coefficients in Equation 4. For Dense, select every routed expert and choose the positive uniform integer rank minimizing the absolute parameter-budget mismatch; report the remaining mismatch. The recorded ranks are 7 for VCR and 8 for NLVR2.
4. For Uniform-layer TOP32, replace K_l with $\lfloor (l+1)S/L \rfloor - \lfloor lS/L \rfloor$ using zero-based l. Select the highest-scoring experts under these quotas, with expert-ID tie-breaking. Keep rank 32 and the original total slot count.
5. For TOP-low-rank, copy the original TOP IDs exactly and use the task's Dense rank. Set $\alpha = 2r$ and recompute the complete parameter count, including DoRA magnitudes. Do not label this comparison budget-matched.
6. Save identities, counts, ranks and budget totals before supervised adaptation. Apply adapters only to the designated routed-expert projections; all pretrained experts remain available to native forward routing. Allocation metadata alone do not verify optimizer rebinding or actual training exposure.

Tie-breaking and random-number implementations can change regenerated expert IDs. The uniformly redistributed follow-up uses the recorded expert-ID tie-break; all historical comparisons must use their saved manifests rather than an assumed equivalent regeneration. The pseudocode assumes valid normalized per-layer profiles; a layer with no valid calibration mass requires a new documented profiling decision.

Table 15: What the anonymous supplement supports.

Operation	Available evidence and scope
Rescore saved answers	Raw-output records, label/group keys and strict parsing entry points
Recompute reported statistics	Frozen correctness inputs, resampling implementations and numerical reference outputs
Inspect allocation and budgets	Saved expert lists, sanitized configs/runtime manifests and the allocation specification above
Replay checkpoint inference	Unavailable: trained adapters, training shards and merged weights are not retained
Replay exact historical training	Unavailable from this package: complete runtime state and per-sample exposure traces are not preserved

The standalone supplement guide `METHOD_ALLOCATION.md` maps these steps to the packaged records. It does not provide new experimental predictions or claim recovery of missing weights.