Auditing Neural Thickets with Low-Rank Routes

Miroslav Lžičař [1]

Abstract

Neural thickets suggest that many task experts lie dense around pretrained weights, but density alone does not imply that such experts are easy to discover. We audit whether those experts can be found through much smaller learned low-rank routes under deployable selection and matched total query budgets. Here, "rank" denotes the dimension of a linear search basis inside an 8-dimensional master perturbation span. On a four-window Countdown audit with Qwen2.5-7B-Instruct, strict selected-before-test accounting gives a mixed and selector-sensitive picture: fixed low-rank controls beat selected dense in two windows and fall below it in two others, while exact-test-oracle diagnostics are stronger but non-deployable. Independent-noise and one-window random-abundance checks preserve the same caution. The result is negative for a unique reward-learned core or one-shot query-efficiency claim, but positive for low-rank routes as probes of compressible, degenerate, and selector-sensitive neighborhoods around frozen checkpoints.

1. Introduction

Pretrained language models appear to occupy weight-space neighborhoods that contain many nearby task-improving experts. Recent work on neural thickets sharpens this density claim (Gan & Isola, 2026), but density alone does not imply discoverability. We audit whether small learned low-rank search routes can expose useful local experts around a fixed checkpoint under deployable selection and matched total query budgets. In our black-box setting, a pilot pool of weight perturbations is scored, tiny learned or control bases are constructed inside the same master span, and downstream local search is restricted to those bases.

This research was conducted independently, without the use of company resources. [1]Deep MedChem. Correspondence to: Miroslav Lžičař <lzicar2000@gmail.com>.

Workshop on Weight-Space Symmetries, held in conjunction with the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

This question is adjacent to, but distinct from, work on low-dimensional adaptation and weight-space geometry. Intrinsic-dimension and PEFT results show that useful updates often lie in low-dimensional or low-rank parameterizations (Li et al., 2018; Aghajanyan et al., 2021; Hu et al., 2022; Lester et al., 2021; Gamal Abdel Hameed et al., 2024). Randomized subspace methods, random-embedding Bayesian optimization, and activation-guided zeroth-order optimization further suggest that structured perturbation spaces can be a practical alternative to isotropic search (Wang et al., 2016; Letham et al., 2020; Chen et al., 2025; Lin et al., 2026). SPSA, NES, and MeZO-style LLM fine-tuning are also relevant black-box or zeroth-order baselines, but they use perturbations to optimize updates rather than to audit pre-existing local experts (Spall, 1992; Wierstra et al., 2014; Malladi et al., 2023). Weight-space connectivity, model-merging, task arithmetic, and universal-subspace studies likewise suggest that nearby solutions can share accessible reusable structure (Garipov et al., 2018; Wortsman et al., 2022; Ilharco et al., 2023; Ainsworth et al., 2023; Kaushik et al., 2025). Unlike these lines of work, we do not fine-tune with gradients, estimate a ZO update, or average trained checkpoints; instead, we test whether black-box local perturbations reveal small searchable routes around a single frozen model. The WSS connection is an empirical weight-space-learning one: approximate degeneracy would appear as multiple small routes with similar behavior, not as a proven symmetry orbit. The object of study is therefore not a new tensor factorization method, but an empirical low-rank search structure: can a tiny route in parameter space render a dense local thicket searchable under fair accounting?

Our claim is intentionally narrow. On Countdown (Gandhi et al., 2024; Jiayi Pan, 2024), a four-window Qwen2.5-7B-Instruct (Qwen, 2024) audit shows that strict selected-by-evaluation accounting turns the earlier apparent low-rank gains into a window-dependent diagnostic. In the two original windows, the best fixed low-rank route is below base and only sometimes edges selected dense. In two fresh windows, low-rank controls can exceed selected dense, but the strongest route switches among residual, random, and bagged controls and still does not yield a uniform deployable advantage. Pilot-oracle and exact-test-oracle diagnostics are stronger, but they are not deployable selection rules. The

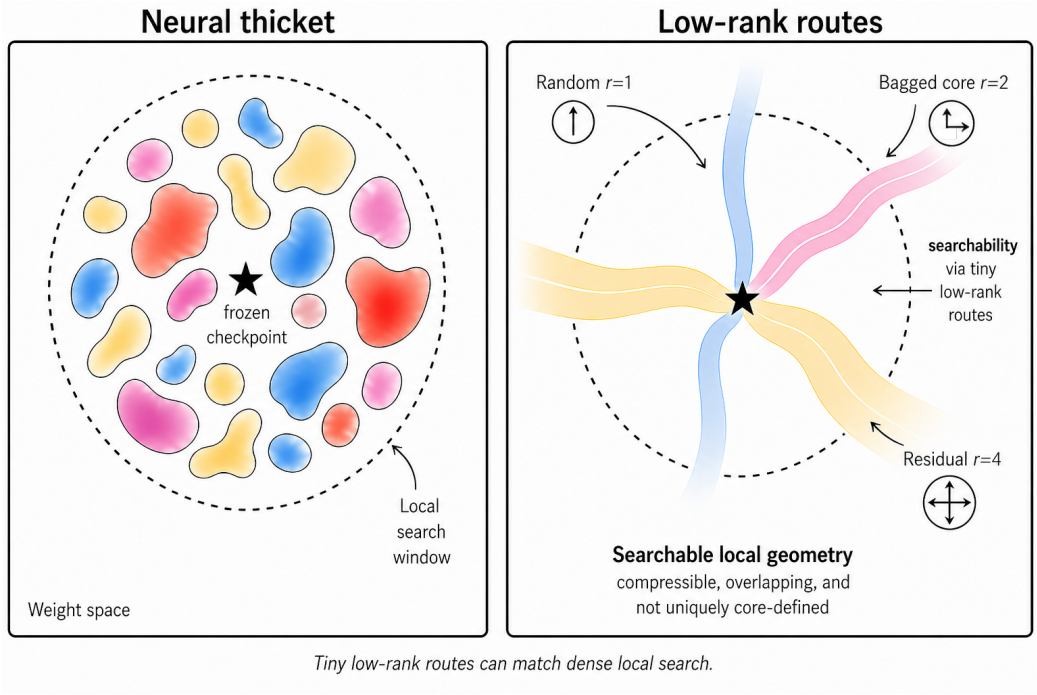

Figure 1. **Auditing neural thickets with low-rank routes.** Left: neural thickets posit many task-improving perturbations near a frozen checkpoint within a local weight-space window. Right: downstream black-box search is restricted to tiny routes inside a fixed master perturbation span, shown by the pre-specified random $r{=}1$, bagged-core $r{=}2$, and residual $r{=}4$ candidates. In held-out Countdown windows, selected-by-evaluation rows are weaker and more window-dependent than exact-test-oracle diagnostics; the supported geometry is compressible and overlapping, but not uniquely explained by a learned core or by a deployable selection advantage.

supported interpretation is negative for a unique reward-learned core or one-shot query-efficiency claim, but positive for local low-rank routes as probes of compressible, degenerate, and selector-sensitive neighborhoods. The main text focuses on this Countdown audit; broader scale and benchmark-audit caveats are presented in the appendix.

2. Method

Notation and search unit. Let θ_0 denote the frozen checkpoint and let $u_1, \ldots, u_8$ be deterministic layer-tied seeded Gaussian perturbation templates over eligible non-visual model parameters. Equal-shaped parameter tensors share the same seed replay pattern, so these are structured local directions rather than independent full-parameter Gaussian draws. We retain this stream for compatibility with the pre-specified audit; Table 2 and the artifact report a matched parameter-name stream check with the same qualitative caution. A candidate in the master span is $\theta_0 + \sigma \sum_{j=1}^{8} c_j u_j$. A rank-$r$ route is an r-dimensional linear subspace of this coefficient space; in this paper, "rank" therefore refers to the search-basis dimension, not to LoRA matrix rank. We use FPE (forward proposal evaluations) for the number of downstream proposals scored inside a single search family: FPE$=16$ means that sixteen candidate perturbations are proposed, scored on the evaluation move and select slices, and selected before the disjoint exact-test slice is read. Exact score is arithmetic correctness on the held-out test slice; the total budget adds the 32 pilot evaluations to downstream search when comparing against dense matched-total baselines.

Two-stage pipeline. We use a two-stage black-box pipeline around Qwen2.5 checkpoints (Qwen, 2024), with the main result reported at Qwen2.5-7B-Instruct. First, we sample a pilot pool of perturbations inside the master span and score them on pilot move and select splits. The resulting pilot coefficients and rewards define learned candidate bases such as reward ridge, reward PLS, and the bagged-core estimator (Hoerl & Kennard, 1970; Wold et al., 1984); the same master span also defines random and residual rank-matched controls. Second, we run local black-box search inside each route, select candidates on a held-out evaluation split, and report exact test performance.

Rank-matched controls. The central control is *rank matching*. For every learned rank-r basis, we compare against: (i) full-span dense search in the 8-dimensional master span, (ii) rank-matched random search, and (iii) rank-matched residual search inside an r-dimensional slice of the orthogonal complement of the bagged-core basis. Because residual search uses the complement of the bagged-core ba-

Algorithm 1: Low-rank route search

1. Sample a pilot pool of perturbations in an 8-dimensional master span.
2. Score pilot perturbations on pilot move/select slices.
3. Fit or construct a tiny route (learned, random, or residual control).
4. Calibrate each family to the same KL target.
5. Run local black-box search only inside the chosen basis.
6. Select candidates on the evaluation slice and report exact test performance; keep exact-test-oracle scores only as diagnostics.

Figure 2. Compact view of the two-stage pipeline.

sis, a residual row that matches or exceeds the learned bases is not independent of the pilot stage; nonetheless, it provides direct evidence against the uniqueness of the bagged-core or reward basis. Rank matching removes a confound in our earlier protocol, in which complement controls implicitly searched a larger space than the learned basis. The main text reports the fixed candidates from the 7B confirmation, while the appendix presents the broader rank-matched control study.

Protocol and reported metrics. The confirmation uses the fixed candidate set {random rank 1, bagged-core rank 2, residual rank 4}, downstream budgets $\{4, 8, 16\}$, a pilot pool of size 32 drawn from a maximum pool of 64, ten low-rank runs per row, and a KL target of 0.02 calibrated from two proposal samples on a small fast-evaluation split. Generation is deterministic, with temperature 0, sampling seed 42, and a Countdown maximum generation length of 48. Because the public Countdown release used here exposes only a single train split, we define fixed non-overlapping experimental slices within it: pilot move `0:16`, pilot select `16:24`, evaluation move `24:40`, evaluation select `40:56`, and exact-test windows `64:128`, `128:192`, `192:256`, and `256:320`. The main table reports raw exact successes out of 64 examples. Deployable dense and low-rank rows use the selected-by-evaluation candidate; pilot and pilot+dense oracle columns are exact-test diagnostics. No low-rank-vs-dense difference in this audit should be read as statistically significant. Reproduction commands and cached selected-run summaries are provided in the supplementary material.

3. Experiments

Case study and metric. We focus the main text on Countdown, the case study in which our total-budget accounting is most direct and on which we ran the held-out confirmation. Countdown uses shaped rewards on the pilot and selection splits, but the reported metric is exact arithmetic correctness on held-out test slices. The confirmation uses Qwen2.5-7B-Instruct on four non-overlapping 64-example windows, with a 32-candidate pilot pool, dense matched-total baselines, and three pre-specified low-rank candidates: random rank 1, bagged-core rank 2, and residual rank 4. These candidates were fixed before held-out evaluation from an earlier candidate-selection screen; all claim-bearing results are the held-out windows in Table 1.

Baselines. *Dense matched-total* runs full-span dense search with the same total number of candidate evaluations as the pilot-plus-search rows and selects candidates on the evaluation slice. *Pilot selected* chooses the pilot candidate by the pilot/evaluation selection score and then reports exact test score. *Pilot oracle* and *pilot+dense oracle* are non-deployable diagnostics that select by exact test score. *Random*, *bagged-core*, and *residual* are the three fixed low-rank candidates chosen before the held-out evaluation and selected by the evaluation slice in the main table. We treat learned bases as probes of local geometry, not as a proposed method to deploy.

Main result. Table 1 supports a more cautious audit than the oracle read. With selected-by-evaluation candidates, the best fixed low-rank row is below selected dense in windows 64–127 and 256–319 at budget $16/48$, but above selected dense in windows 128–191 and 192–255. The strongest route also changes with the window: residual rank 4 is strongest in window 192–255, random rank 1 in window 128–191, and bagged-core rank 2 in window 256–319. The finding is therefore not a uniform efficiency claim; it is evidence that low-rank routes expose selector-sensitive local structure whose apparent strength depends strongly on the held-out slice and on whether exact-test oracle selection is allowed.

Pilot accounting. The pilot baselines are essential to the interpretation. Pilot selection is unstable across the four windows, while pilot oracle and pilot+dense oracle reach $14/64$, $9/64$, $13/64$, and $14/64$. These oracle rows explain the earlier optimistic read but are not deployable. The supported claim is consequently narrower than "low-rank beats every fair baseline." We retain selected dense, pilot selected, and oracle diagnostics together in the main table because separating them would understate the accounting subtleties.

4. Discussion

The Countdown evidence supports a limited diagnostic role for tiny low-rank routes, but it does not establish a deployable low-rank improvement over base or dense search. This is the main negative result of the audit: the learned-core

Table 1. Held-out Qwen2.5-7B Countdown confirmation under strict total-budget accounting after the 2026-05-06 direct-vLLM audit. Entries are raw exact successes out of 64. Deployable dense and low-rank rows use selected-by-evaluation candidates; oracle columns are exact-test diagnostics and are non-deployable. Low-rank rows report mean ± sample standard deviation over 10 runs. We make no significance claim.

Window	Budget	Base	Dense sel.	Pilot sel.	Pilot oracle	P+d oracle	Random $r{=}1$	Bagged $r{=}2$	Residual $r{=}4$
64–127	8/40	13	14	10	14	14	11.3±0.9	10.5±1.2	11.9±1.1
64–127	16/48	13	13	10	14	14	10.7±1.3	10.2±1.0	11.7±0.9
128–191	8/40	9	7	6	9	9	**8.0±1.2**	6.2±0.9	6.3±0.8
128–191	16/48	9	7	6	9	9	**7.5±1.2**	6.2±0.8	6.7±1.3
192–255	8/40	9	8	10	13	13	10.8±1.9	11.1±2.0	**11.2±2.7**
192–255	16/48	9	10	10	13	13	9.9±2.3	11.3±2.3	**11.5±2.5**
256–319	8/40	9	8	8	14	14	8.7±1.8	**9.8±1.2**	8.9±1.4
256–319	16/48	9	11	8	14	14	8.5±2.2	**9.5±1.2**	9.0±1.6

and one-shot query-efficiency hypotheses do not survive strict accounting. It is nonetheless a useful refinement of the neural-thickets account, since it shows that dense local expert neighborhoods can look quite different under selected-by-evaluation and exact-test-oracle accounting. The independent-noise check preserves the same qualitative picture, and the one-window random-abundance check shows that useful random rank-1 routes exist but are not reliably selected.

The principal caveat concerns total-budget and selection accounting. Tables 1 and 2 show that strict accounting changes the conclusion: deployable low-rank rows are mixed, oracle diagnostics are stronger but non-deployable, and the strongest selected routes are random, residual, or bagged controls depending on the window. A separate MBPP code-generation audit (Austin et al., 2021) delivers the same caution in a negative direction: after correcting split handling, the best learned rank-1 rows reach only 0.5917 exact, below the MBPP base score of 0.6250, while the best pilot-only candidate and matched-total-budget dense both reach 0.6667. We therefore advance a deliberately modest claim: local neighborhoods around frozen checkpoints can be probed by tiny routes and may be compressible and degenerate, but the present evidence does not establish a unique reward-learned core or a one-shot query-efficiency advantage. For WSS, this is an empirical weight-space-learning audit of approximate degeneracy, not a symmetry-identification method.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, 2021.

Ainsworth, S., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., and Sutton, C. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Chen, Y., Zhang, Y., Liu, Y., Yuan, K., and Wen, Z. A memory efficient randomized subspace optimization method for training large language models. *arXiv preprint arXiv:2502.07222*, 2025.

Gamal Abdel Hameed, M., Milios, A., Reddy, S., and Rabusseau, G. Rosa: Random subspace adaptation for efficient fine-tuning. *arXiv preprint arXiv:2407.07802*, 2024.

Gan, Y. and Isola, P. Neural thickets: Diverse task experts are dense around pretrained weights. *arXiv preprint arXiv:2603.12228*, 2026.

Gandhi, K., Lee, D., Grand, G., Liu, M., Cheng, W., Sharma, A., and Goodman, N. D. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1): 55–67, 1970.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Jiayi Pan. Countdown-tasks-3to4 dataset. Hugging Face dataset, 2024. URL `https://huggingface.co/datasets/Jiayi-Pan/Countdown-Tasks-3to4`.

Kaushik, P., Chaudhari, S., Vaidya, A., Chellappa, R., and Yuille, A. The universal weight subspace hypothesis. *arXiv preprint arXiv:2512.05117*, 2025.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Letham, B., Calandra, R., Rai, A., and Bakshy, E. Re-examining linear embeddings for high-dimensional bayesian optimization. *arXiv preprint arXiv:2001.11659*, 2020.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Lin, W., Jiang, Y., Song, Q., Xiang, Q., and Xu, H. Agzo: Activation-guided zeroth-order optimization for llm fine-tuning. *arXiv preprint arXiv:2601.17261*, 2026.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023.

Qwen. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. *Journal of Machine Learning Research*, 15(27):949–980, 2014.

Wold, S., Ruhe, A., Wold, H., and Dunn, W. J. The collinearity problem in linear regression: The partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23965–23998, 2022.

Table 2. Audit and pilot-accounting summaries that bound stronger efficiency interpretations. Countdown rows charge the same 32-point pilot pool used in the confirmation; MBPP rows come from the corrected split-respecting holdout rerun.

	7B Countdown 64–127 / 128–191	7B Countdown 192–255 / 256–319	MBPP holdout audit
Dense matched-total	selected: 13/64; 7/64	selected: 10/64; 11/64	0.6667 at total budget 36
Strong pilot baseline	pilot selected: 10/64; 6/64; oracle: 14/64; 9/64	pilot selected: 10/64; 8/64; oracle: 13/64; 14/64	best pilot-only: 0.6667
Best low-rank candidate	residual r=4: 11.7 ± 0.9/64; random r=1: 7.5 ± 1.2/64	residual r=4: 11.5 ± 2.5/64; bagged r=2: 9.5 ± 1.2/64	reward ridge / bagged core: 0.5917
Read	below base; only the second selected low-rank row edges selected dense	fresh windows show low-rank selected rows can exceed dense, but not uniformly	corrected split handling removes the earlier broader positive interpretation

A. Appendix

Protocol details. The Countdown confirmation uses Qwen2.5-7B-Instruct with master rank 8, a 32-candidate pilot pool truncated from a maximum pool of 64, downstream budgets $\{4, 8, 16\}$, and ten runs for each reported low-rank control. Perturbations are deterministic layer-tied seeded Gaussian templates over eligible model parameters, calibrated to a common last-token KL target; decoding uses temperature 0, seed 42, and maximum length 48. Because the public Countdown release used here exposes a single train split, we define fixed non-overlapping slices within it: pilot move `0:16`, pilot select `16:24`, evaluation move `24:40`, evaluation select `40:56`, and exact-test windows `64:128`, `128:192`, `192:256`, and `256:320`. Earlier rank-matched studies on Qwen2.5-3B and Qwen2.5-1.5B used the same slice convention and are presented as scale context rather than as the main claim.

Assets and access. The camera-ready code repository will be hosted at `https://github.com/mireklzicar/neural-thickets`. The artifact identifies upstream assets rather than redistributing them: `Qwen/Qwen2.5-7B-Instruct`, `Jiayi-Pan/Countdown-Tasks-3to4`, and the MBPP Hugging Face mirror `google-research-datasets/mbpp` used for the holdout audit. At audit time, the Qwen model card lists Apache-2.0 and is not gated, MBPP lists CC BY 4.0, and the Countdown card is public and non-gated but does not declare an explicit license tag in card/API metadata. Reproduction therefore depends on the current upstream model and dataset cards, licenses, and access terms.

Audit summary. Table 2 collects the caveats that keep the paper's claim narrow. On the four held-out 7B Countdown windows, selected-by-evaluation low-rank routes are weaker and more window-dependent than the earlier exact-test-oracle read. At budget $16/48$, the best low-rank row is below selected dense in windows 64–127 and 256–319, and above selected dense in windows 128–191 and 192–255. On MBPP, the corrected split-respecting holdout rerun no longer supports the earlier broader positive interpretation: the best learned rank-1 rows reach only 0.5917, and reward PLS rank 1 drops to 0.4750.

GPU audit follow-ups. The direct-vLLM rerun did not exactly reproduce the earlier frozen paper table: 45 of 64 compared scalar fields moved outside a tight rounding tolerance, with a maximum change of two exact successes. The revised table therefore uses the direct-vLLM audit outputs, and the pre-audit frozen table is archived in the artifact. We also ran two review-requested checks. First, the matched independent per-parameter noise ablation preserves the qualitative conclusion: in window 64–127, the best low-rank control is residual r=4 at 12.6 ± 1.4/64 versus selected dense at $13/64$; in window 128–191, random r=1 reaches 8.5 ± 1.5/64 versus selected dense at $7/64$ but remains below base at $9/64$. Second, a one-window abundance audit on 64–127 finds many random rank-1 routes with nonzero utility, but not a reliable selected advantage: the dense selected distribution over 10 runs has mean 13.0 ± 0.7/64, while 80 random rank-1 routes have mean 11.0 ± 2.4/64, median $11/64$, range 6–16, and win rates of 15.0% against base and 13.8% against the paired dense-selected run.

Why the claim remains narrow. The 7B confirmation does not pass as a deployable low-rank advantage. Bagged-core rank 2 exceeds selected dense in one fresh window but falls below selected dense in the other three windows, and none of the selected low-rank rows uniformly exceeds the base or the oracle diagnostics. We therefore describe the local

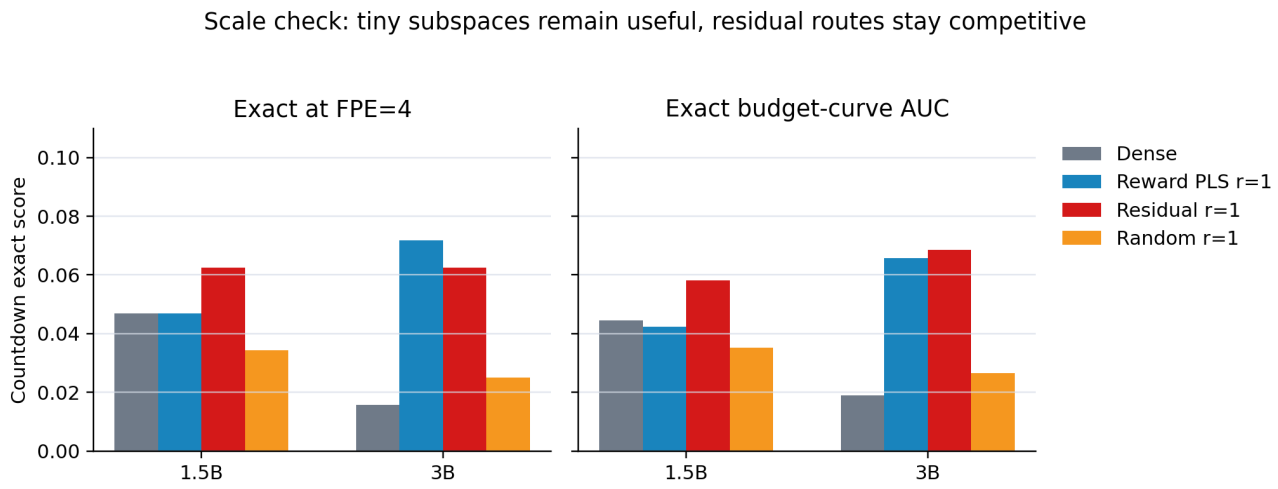

Figure 3. Countdown scale check on Qwen2.5-1.5B and Qwen2.5-3B. Reward PLS rank 1 can be useful, but residual controls are competitive or stronger; the confirmed 7B selected-candidate audit is likewise not a learned-basis advantage.

geometry as selector-sensitive and potentially compressible or degenerate, rather than as uniquely core-contained or uniquely reward-learned.

Scale checks. The full scale study provides context rather than a stronger learned-basis claim. On Qwen2.5-1.5B, reward PLS rank 1 shows only a marginal diagnostic edge over dense matched-total at the largest total budget. On Qwen2.5-3B, residual rank 1 outperforms the learned rows. On Qwen2.5-7B, the held-out selected-candidate audit weakens the apparent low-rank gains. The scale checks therefore support the cautious geometric claim of this paper: small useful routes can appear in diagnostics, but the highest-scoring route is model- and window-dependent rather than uniquely explained by the learned core.

Amortized-transfer probe. We tested whether the original learned Countdown basis improves under amortization, in which its pilot cost is shared across multiple held-out target windows. Across three non-overlapping windows, transferred reward PLS rank 1 averages 0.0642 exact, below dense downstream search at 0.0712, residual rank 1 at 0.0781, and transferred pilot selection at 0.0833. The best transferred pilot perturbation in each window averages 0.1042. Amortization therefore does not establish a strong efficiency claim; instead, it reinforces the observation that the pilot pool itself contains reusable local experts.

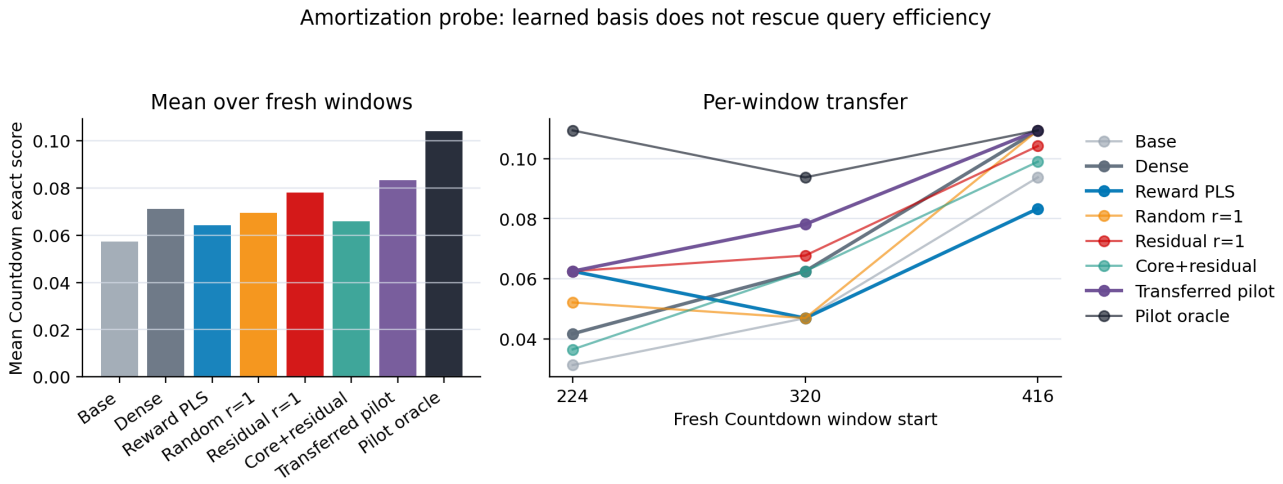

Figure 4. Amortized-transfer probe on three held-out Countdown windows. Reusing the reward-PLS basis without refitting stays above the base score but does not exceed dense search, residual search, or transferred pilot selection on average.

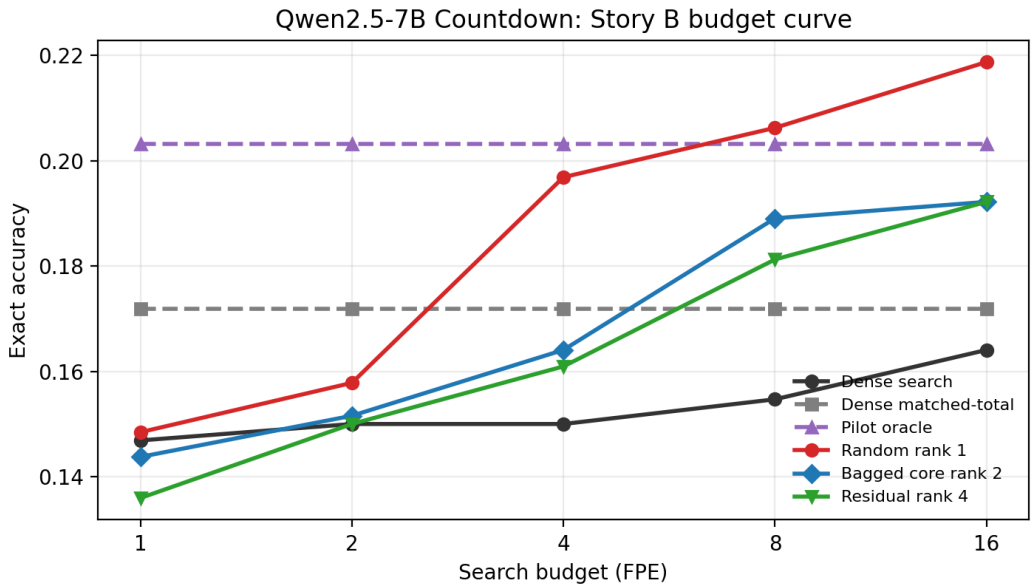

Figure 5. Historical Qwen2.5-7B Countdown budget curve used only to choose fixed candidates for the held-out confirmation. This figure is not the claim-bearing result. Dense matched-total and pilot oracle are shown as accounting baselines; random and residual controls are competitive with learned bases, motivating the control-dominated selected-candidate audit in Table 1.

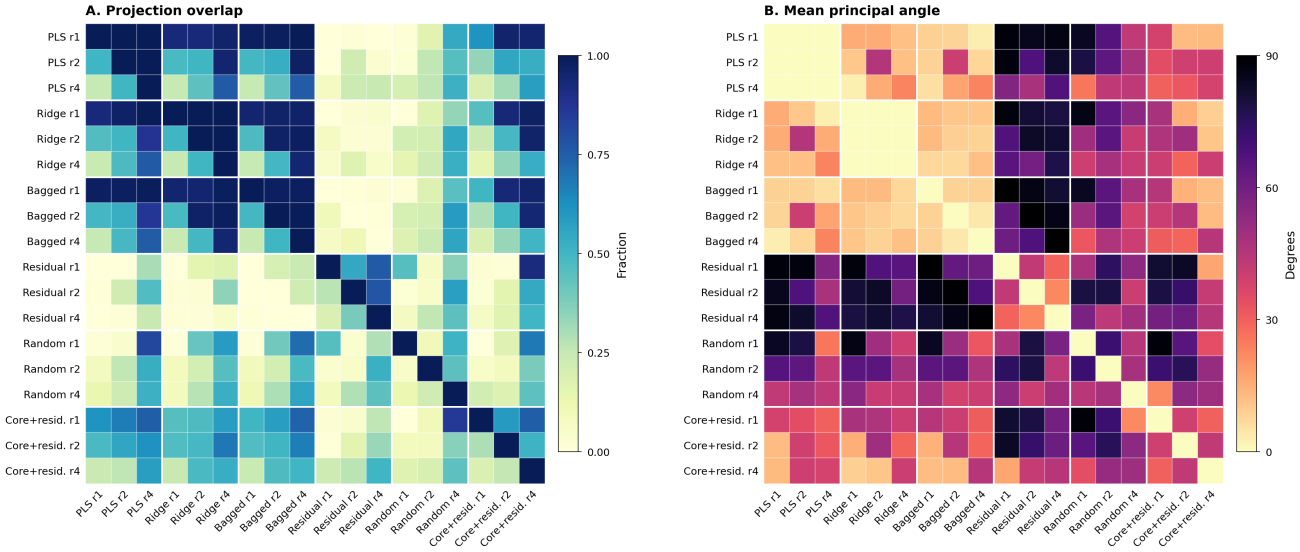

Figure 6. Countdown subspace-overlap diagnostic across the learned and control families. Reward-informed bases show nontrivial alignment with one another, but the residual and mixed controls remain geometrically distinct rather than collapsing onto a single shared low-rank core. This figure is therefore consistent with our main claim that the local neighborhood is searchable and compressible, but not uniquely core-contained.