SPARSE NEURAL THICKETS: SAMPLING THE SMALL WEIGHTS SUBSPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

The loss landscape around pretrained models contains dense neighborhoods of task-improving solutions, known as *Neural Thickets*. We show that this structure persists even when perturbations are restricted to low-dimensional parameter subspaces, revealing *Sparse Neural Thickets*. Strikingly, subspaces of low-magnitude parameters are consistently enriched with high-performing solutions compared to subspaces of large-magnitude parameters and dense subspaces. This suggests that favorable regions of the local loss landscape are concentrated in an easily identifiable subset of parameters. Leveraging this structure, we introduce **Su**bspace **P**arameter-**E**fficient **S**ampling (SuPES). The often higher density of good solutions can enable more efficient exploration: SuPES can achieve higher accuracy at comparable sampling budgets or comparable accuracy with fewer samples, while perturbing substantially fewer parameters than full-space sampling. Experiments across language models, parameter scales, and datasets establish Sparse Neural Thickets and demonstrate their potential for parameter- and sample-efficient sampling around pretrained models.

1 INTRODUCTION

Gan & Isola (2026) provide empirical evidence for the existence of so-called *Neural Thickets*: neighborhoods around pretrained large language models (LLMs) in parameter space in which task-improving solutions are sufficiently dense to be discovered through random sampling. To explicitly construct such solutions, they propose *RandOpt*, which repeatedly samples random perturbations of all parameters of a pretrained model, evaluates the resulting models on the target task, and ensembles the top-k candidates. In addition to exploiting the density of improving solutions, this procedure benefits from the model diversity naturally induced by random sampling.

This observation suggests a compelling alternative approach to adapting large foundation models to downstream tasks. Rather than explicitly estimating a descent direction from gradient information, useful adaptation directions are sufficiently abundant near a pretrained solution that can be discovered through sampling alone. This insight has already inspired several extensions that leverage sampling for (near) zero-order fine-tuning, reducing reliance on computationally expensive and memory-intensive gradient information (Zhang et al., 2026; Venkatkrishna et al., 2026; Kim et al., 2026; Cendra et al., 2026). A better understanding of the mechanisms underlying the success of such methods may therefore both reveal properties of the local loss landscape around pretrained models and inform the design of more efficient sampling-based adaptation algorithms.

A central question remains, however: *How much of the parameter space is actually required for Neural Thickets to emerge?* RandOpt perturbs all model parameters simultaneously, implicitly searching in the full-dimensional parameter space. Yet successful adaptation of pretrained models is often highly structured and parameter-efficient fine-tuning (PEFT) methods suggest that modifying only a small fraction of the available degrees of freedom should be required (Hu et al., 2021; Liu et al., 2024; Xu et al., 2023; Wu et al., 2024). This raises the possibility that the abundance of task-improving solutions observed by Gan & Isola (2026) does not fundamentally depend on access to the full parameter space. Instead, similarly dense sets of solutions may persist in substantially lower-dimensional, sparse parameter subspaces.

In this work, we show that this is indeed the case. We find that Neural Thickets persist even when random perturbations are restricted to only a small fraction of model parameters. We refer to this

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.

phenomenon as *Sparse Neural Thickets*. Remarkably, perturbing as little as 1% of the parameters can yield sampling performance competitive with perturbing the full model. These results suggest that the local abundance of task-improving solutions is not merely a consequence of searching an extremely high-dimensional space. Rather, useful adaptation directions remain abundant under severe restrictions of the searchable parameter space. This restriction could also allow for a more targeted sampling and better coverage of potential solutions.

This intuition naturally raises a second question: *Are all sparse parameter subspaces equally rich in task-improving solutions?* Recent evidence suggests that parameter magnitude provides a meaningful signal for identifying parameters relevant to adaptation. In particular, low-magnitude parameters have been associated with large gradient magnitudes (Zhou et al., 2025), suggesting that they may play a disproportionate role when adapting pretrained LLMs to new tasks. Furthermore, concentrating adaptation on this parameter subset that appears to be less important poses a lower risk of catastrophic forgetting of the pre-training task. This motivates studying parameter magnitude not merely as a pruning criterion, but as a potential indicator of where task-improving perturbations are concentrated in the local loss landscape.

We therefore compare sparse subspaces induced by parameter magnitude and find that subspaces consisting of low-magnitude parameters are systematically enriched in useful solutions relative to corresponding high-magnitude subspaces. This result provides a complementary perspective on the role of low-magnitude parameters in model adaptation: Not only can these parameters exhibit strong gradient-based adaptation signals, but the subspaces they induce also contain a greater density of solutions accessible through random perturbation. More broadly, our findings indicate that Neural Thickets possess non-trivial internal structure—their solution density depends on *which* parameters are available for perturbation, rather than only on the dimensionality of the search space.

Beyond their implications for the geometry of the post-training loss landscape, Sparse Neural Thickets also suggest opportunities for making sampling-based adaptation more efficient. Restricting perturbations to a small parameter subset reduces the number of parameters that must be modified and stored, while the concentration of useful solutions in particular subspaces can improve the effectiveness of a fixed sampling budget. Furthermore, the existence of more high-quality solutions could also reduce our need for a high number of ensemble members of the final solution, leading to computational savings at inference.

However, sparsification changes the appropriate scale of random perturbations, which is critical for the density of solutions in sampled models. We characterize this relationship empirically and derive a practical heuristic for adapting the perturbation scale as the size of the perturbed parameter subset changes. Building on these observations, we introduce *SuPES*, a sparse sampling approach that restricts perturbations to selected parameter subsets and achieves competitive performance with RandOpt, while frequently requiring fewer samples or fewer models in the resulting ensemble.

Contributions.

- **Sparse Neural Thicket existence.** We provide empirical evidence for *Sparse Neural Thickets*: dense neighborhoods of task-improving solutions that persist under severe restrictions of the searchable parameter space, including settings in which only 1% of the model parameters are perturbed.
- **Enrichment of low-magnitude subspaces.** We show that sparse subspaces induced by low-magnitude parameters are enriched in task-improving solutions relative to corresponding high-magnitude parameter subspaces, revealing structure in the distribution of useful adaptation directions around pretrained models.
- **Perturbation-scale scaling.** We show that sampling performance depends critically on the perturbation scale and characterize how an effective scale changes with the number of perturbed parameters. Based on this relationship, we derive a practical heuristic for transferring perturbation scales across different sparsity levels.
- **Sparse sampling algorithm.** Building on these findings, we introduce *SuPES*, a parameter-efficient variant of RandOpt that restricts sampling to selected parameter subspaces. SuPES achieves competitive performance with full-parameter RandOpt and, depending on the task, can improve sample or model efficiency.

2 Related Work

Loss landscapes around pretrained models. Recent works have shown that pretrained neural networks lie in locally flat, basin-like loss landscapes (Keskar et al., 2017; Chen et al., 2026). Gan & Isola (2026) demonstrated that this flat minimum contains *thickets* of task-specific solutions, which can be exploited using random sampling. Further works have emphasized that large pretrained models have an intrinsic low-dimensional representation (Aghajanyan et al., 2020) which can be exploited during fine-tuning. Loss landscapes around pretrained LLMs are observed to have low-dimensional curvature (Liang et al., 2026), meaning that zero-order fine-tuning approaches are dominated by a few high-curvature dimensions. If adaptation is possible in low-dimensional spaces, then high-performing perturbations may also exist there. Yet, the role of low-dimensional subspaces for *Neural Thickets* remains unexplored.

Zero-order optimization. Zero-order optimization (Liu et al., 2020) tries to optimize functions without explicit gradient computation. Since gradient computation is expensive for LLMs, recent works argued that this kind of optimization algorithm is underexplored in the deep learning context (Liu et al., 2026). Recent advancements in applying evolutionary style algorithms (Zhang et al., 2026; Qiu et al., 2026) have demonstrated the power of zero-order optimization algorithms. Further zero-order optimization methods were developed for training LLMs (Malladi et al., 2024; Zhang et al., 2024), exhibiting the power of zero-order optimization in deep learning. Despite these advances, most zero-order methods rely on gradient approximation and iterative steps. In contrast, our approach relies solely on fully parallelizable random sampling in the weight space.

Sparse training. As most deep learning methods operate on the whole, largely overparameterized parameter space (with the number of parameters in LLMs ranging into the billions), there is a strong need for parameter-efficient fine-tuning methods (Han et al., 2024; Wang et al., 2025). Recent works have shown that fine-tuning (Zhang et al., 2023; Song et al., 2024; Morris et al., 2026) and even training (Frankle & Carbin, 2019; Li et al., 2018; Gressmann et al., 2020; Burkholz, 2024; Gadhikar et al., 2025; Jacobs et al., 2026a;b) are feasible in small parameter subspaces, as proven by theoretical work on lottery ticket existence (Orseau et al., 2020; Pensia et al., 2020; Burkholz, 2022b; da Cunha et al., 2022; Burkholz, 2022a; Burkholz et al., 2022; Fischer & Burkholz, 2021; Fischer et al., 2021; Gadhikar et al., 2023). Modern LoRA-style algorithms (Hu et al., 2021; Xu et al., 2023; Liu et al., 2024) exploit these tiny subspaces by forcing task-specific adapters to be low rank. Zhou et al. (2025) furthermore identified those subspaces a priori based on the absolute magnitude of parameters.

3 Methods

Our goal is to understand whether the local abundance of useful models, as shown by Gan & Isola (2026), depends on searching the full parameter space. We therefore restrict random perturbations to coordinate subspaces and study how the prevalence of favorable solutions changes. We first introduce the solution-density criterion and the enrichment measure used to compare search spaces, then define the sparse parameter subspaces considered in this work and the corresponding SuPES sampling procedure. Finally, because restricting the number of perturbed coordinates changes the typical size of a perturbation, we derive a simple rule for adapting the perturbation scale across subspace dimensions.

Solution Density and Enrichment We quantify the prevalence of favorable perturbations using the solution density introduced by Gan & Isola (2026).

Definition 1 (Solution Density (Gan & Isola, 2026))**.** Let $s : \mathbb{R}^d \to \mathbb{R}$ denote the performance metric for model parameters $\theta \in \mathbb{R}^d$. The solution density $\delta(m)$ is the probability that a random perturbation improves the base model's score by at least a margin m:

$$\delta(m) = P_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)} \left[s(\theta + \epsilon) \geq s(\theta) + m \right]. \tag{1}$$

Here, σ determines the scale of the local Gaussian neighborhood.

The solution density can be interpreted as the probability that a randomly sampled perturbation reaches a given performance level relative to the base model. A perturbation with $m > 0$ is *task-*

improving, while $m = 0$ corresponds to a *task-preserving* perturbation. Evaluating the density at different values of m therefore characterizes different parts of the local performance distribution.

For perturbations restricted to a subspace $\mathbb{S}_p$, we estimate the corresponding solution density from N independent samples as

$$\hat{\delta}_p(m) = \frac{1}{N} \sum_{\epsilon} \mathbf{1} \left[s(\theta + \epsilon) \geq s(\theta) + m \right], \tag{2}$$

where the samples are drawn according to the restricted perturbation scheme defined below. This estimator allows us to compare the prevalence of solutions of a given quality across parameter subspaces, perturbation fractions, and noise scales.

To quantify relative differences in solution density, we define the enrichment ratio between two perturbation strategies or subspaces p and q as

$$r(p, q) = \frac{\hat{\delta}_p(m)}{\hat{\delta}_q(m)}. \tag{3}$$

A ratio $r(p, q) > 1$ indicates *enrichment* of p relative to q, whereas $r(p, q) < 1$ indicates *depletion*. Thus, enrichment means that perturbations exceeding threshold m occur more frequently under strategy p than under strategy q. The perturbation-scale protocol used for each comparison is specified with the corresponding experiment.

Low-magnitude parameter subspace We distribute the perturbations evenly across the layers of the model and select the lowest $p\%$ magnitude parameters per layer. For layer j with parameters θ^j, we calculate a fixed perturbation mask containing indices $\mathbb{A}_p^j = \{i : |\theta_i^j| \leq t_j(p)\}$, where $t_j(p)$ is the threshold corresponding to the p-th percentile of absolute parameter magnitudes ($|\theta_i^j|$) in that layer. With $\mathbb{A}_p = \cup_j \mathbb{A}_p^j$, $\mathbb{S}_p = \text{span}\{e^{(i)} : i \in \mathbb{A}_p\} \subseteq \mathbb{R}^d$ defines the low-magnitude-parameter subspace, where $e^{(i)}$ denotes the i-th standard basis vector of $\mathbb{R}^d$. The random and largest-magnitude parameter subspace baselines are constructed analogously.

Subspace Parameter-Efficient Sampling (SuPES) Following Gan & Isola (2026), we sample Gaussian perturbations $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ and construct perturbed parameters $\theta_{\text{perturb}} = \theta + \epsilon$. SuPES restricts perturbations to a subspace $\mathbb{S}_p$ by masking the perturbation vector before applying it to the model parameters. Restricting these perturbations to the subspace $\mathbb{S}_p$ defined above is achieved by masking the perturbation vector ϵ^j of each layer by a mask vector $m_i^j = \begin{cases} 1, & i \in \mathbb{A}_p^j \\ 0, & i \notin \mathbb{A}_p^j \end{cases}$ and calculate $\theta_{\text{perturb}}^j = \theta^j + (m^j \odot \epsilon^j)$ where $\odot$ denotes elementwise multiplication, and thus $m^j \odot \epsilon^j \in \mathbb{S}_p$. In this way, we only perturb the masked (in our case low-magnitude) parameters.

Norm-preserving heuristic When sampling only low-magnitude parameters, the expected squared Euclidean distance of the perturbed models to our base model in the weight space shrinks compared to full perturbation. Since Neural Thickets are observed in a local neighborhood around a base model, restricting perturbations to fewer parameters using the same σ reduces the radius of exploration during the sampling of the local neighborhood compared to full perturbation. To adjust for this, we need to rescale the noise scale σ appropriately. Preserving the expected squared Euclidean distance provides a simple geometric criterion for sampling perturbations from neighborhoods of comparable size. When perturbing the full model, the perturbations have an expected squared Euclidean distance of $\mathbb{E}[||\epsilon||^2] = d\sigma^2$. Perturbing only a fraction p of the parameters yields an expected squared Euclidean distance of $\mathbb{E}[||\epsilon||^2] = p \cdot d\sigma^2$, hence to achieve similar squared Euclidean distance we need to adjust the noise scale σ by $\sqrt{1/p}$, leading to

$$\sigma_{\text{rescaled}} = \sigma / \sqrt{p}. \tag{4}$$

Our expected squared Euclidean distance is $\mathbb{E}[||\epsilon||^2] = p \cdot d\sigma^2_{\text{rescaled}} = p \cdot d\left(\frac{1}{p}\sigma^2\right) = d\sigma^2$, the same as in the full perturbation case. This gives us a norm-preserving heuristic, assuming that parameter-space distance provides a useful proxy for the exploration radius around the pretrained model.

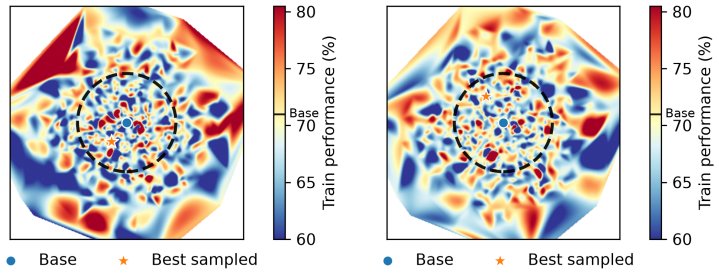


Figure 1: Perturbed weights projected into 2D using random projections. The color indicates the performance on the training data (red is better). Qwen2.5-1.5B-Instruct on GSM8K. (Left) SuPES (1%) with $\sigma = 0.007$. (Right) RandOpt with $\sigma = 0.001$.

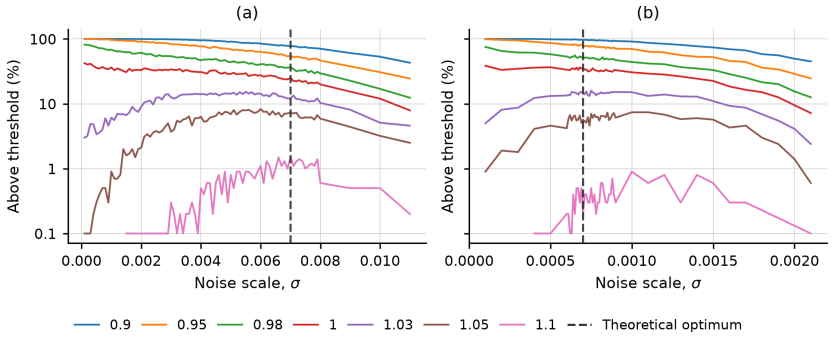


Figure 2: Solution densities for different thresholds across different values of σ. The heuristic reference σ is computed relative to the 1% case. (a) 1% low-magnitude parameters. (b) full perturbation.

4 Results

We empirically examine our claims by evaluating the solution densities around Qwen2.5-1.5B-Instruct (Bai et al., 2023) on GSM8K (Cobbe et al., 2021) and Countdown (Gandhi et al., 2024), with the Countdown results reported in Appendix A.1. For this analysis, we use 110% of the base model's performance as the primary threshold and refer to perturbations exceeding it, i.e., $m = 0.1\, s(\theta)$, as *high-performing*. We further evaluate downstream accuracy using Qwen2.5-1.5B-Instruct and OLMo3-7B (Olmo et al., 2026) on GSM8K, OlympiadBench (He et al., 2024), Countdown, and USPTO50K (van der Lingen, 2023).

4.1 Calibrating the Perturbation Scale

The solution density depends strongly on the perturbation scale σ. Figure 2 shows this dependence for full-space perturbations and for perturbations restricted to the lowest-magnitude 1% of parameters. For very small σ, perturbations remain close to the pretrained model and high-performing solutions are rare. As σ increases, the density of high-performing solutions initially rises before declining again once perturbations move too far from the pretrained solution. Importantly, the location of this high-density region shifts substantially when the perturbation dimension changes.

This shift is qualitatively consistent with the norm-preserving heuristic (Equation 4): Reducing the number of perturbed parameters requires a larger perturbation scale to explore a neighborhood of comparable expected Euclidean radius. The heuristic is purely geometric and does not model the local loss landscape, so we use it as a practical calibration rule rather than as a prediction of the exact optimal scale. In the following experiments, we therefore report results both using empirically favorable values of σ and using the norm-preserving scaling rule. This allows us to separate conclusions that depend on careful scale selection from those that persist under a simple transfer rule.

Perturbation strategy	Density at the 110% threshold (%)	Enrichment compared to full space
Low-magnitude 1%	1.14 [0.9, 1.5]	1.78 [1.18, 2.83]
Random 1%	1.04 [0.8, 1.4]	1.63 [1.05, 2.57]
Full space	0.64 [0.5, 0.9]	1
Large-magnitude 1%	0.60 [0.5, 0.9]	0.56 [0.3, 0.96]

Table 1: Estimated density of perturbations exceeding 110% of the unperturbed model's training reward on GSM8K with Qwen2.5-1.5B-Instruct. Each strategy uses an empirically selected noise scale and 5,000 sampled perturbations. For densities, brackets give 95% Wilson confidence intervals. For enrichment, brackets give 95% bootstrapped confidence intervals, calculated according to appendix A.6.

4.2 Solution Densities and Enrichment

We next ask whether favorable perturbations remain abundant when the search is restricted to a small coordinate subspace. We estimate solution densities using Equation 2 on training reward, since both *RandOpt* and *SuPES* select candidate perturbations according to their training performance.

We use the 110% threshold introduced above as our primary comparison. To determine whether the observed behavior is specific to the high-performing tail, we additionally evaluate thresholds at 90%, 95%, 98%, 100%, 103%, and 105% of the base model's performance. In particular, the 100% threshold corresponds to task-preserving perturbations, while 90% corresponds to $m = -0.1\, s(\theta)$. Complete threshold and perturbation-fraction sweeps are reported in Appendix A.1.

We compare perturbing the lowest-magnitude, randomly selected, and highest-magnitude 1% of parameters with full-space perturbation. As shown in Section 4.1, the relevant noise scale changes substantially with perturbation dimension. We therefore first compare each strategy near an empirically favorable operating point, selecting the perturbation scale separately for each strategy. Table 1 reports the resulting solution densities and enrichment ratios at the primary 110% threshold on GSM8K with Qwen2.5-1.5B-Instruct. This comparison uses 5,000 independent perturbations per strategy; the broader sweeps in A.1 use 1,000 perturbations per setting.

The random control is nearly as dense as the low-magnitude subspace, indicating that reducing the search dimension itself accounts for a substantial part of the observed enrichment. At the same time, dimensionality alone does not explain the full pattern. The large-magnitude subspace is depleted relative to full-space perturbation and has a weaker high-performing tail than either of the other sparse subspaces. Thus, the coordinates defining the restricted search space also influence the density of favorable solutions. Table 1 reports the corresponding uncertainty estimates; bootstrap confidence intervals are computed as described in Appendix A.6.

Figure 1 provides a complementary visualization of these sampled neighborhoods. We randomly project the sampled parameter perturbations into two dimensions and color them by their training reward, following the visualization of Gan & Isola (2026). High-reward perturbations remain interspersed throughout the projected local neighborhood, qualitatively resembling the thicket structure observed under full-space sampling. Together with the density estimates above, this motivates the term *Sparse Neural Thickets*: favorable perturbations remain abundant even when the search is confined to a small coordinate subspace.

The strong dependence on σ raises a second question: how should the perturbation scale change as the search dimension changes? The norm-preserving heuristic from Section 4.1 does not match the empirically optimal σ exactly, but it captures the shift in the relevant scale and therefore provides a useful reference point when searching for a suitable perturbation scale. To test whether the density pattern persists under this simple rule, we start from an empirically favorable scale for the 1% subspace and use the heuristic to obtain scales for larger perturbation fractions. At the 110% threshold, the resulting estimated densities are 1.5% for low-magnitude restriction, 0.6% for random restriction, 0.2% for full-space perturbation, and 0.1% for large-magnitude restriction (Table A.4).

Because the 1% reference scale is selected empirically while the scales at larger fractions are obtained from the heuristic, these numbers should not be interpreted as a comparison between independently optimized operating points. Rather, they show that the qualitative enrichment pattern persists when a calibrated scale is transferred across perturbation dimensions using a simple geometric rule.

Model	Dataset	Base	SuPES (1%)	RandOpt
Qwen2.5-1.5B-Instruct	GSM8K	62	**77.86**	75.66
Qwen2.5-1.5B-Instruct	Countdown	13	38.95	**39.85**
Qwen2.5-1.5B-Instruct	OlympiadBench	14.5	**24.47**	24.26
Qwen2.5-1.5B-Instruct	USPTO50K	7	**31.97**	31.86
OLMo3-7B-Instruct	GSM8K	87.41	**89.007**	88.476
OLMo3-7B-Instruct	Countdown	67.60	90.35	**90.7**
OLMo3-7B-Instruct	OlympiadBench	47.05	**61.392**	58.650

Table 2: Test accuracy (%) of the top-50 ensemble.

Complete results across perturbation fractions, reward thresholds, and noise-scale protocols are reported in Appendix A.5.

Additional density measurements for OLMo3-7B-Instruct are reported in Appendix A.1. Across Countdown, OlympiadBench, and GSM8K, the upper tail remains sensitive to both the task and the noise scale. These results reinforce the role of σ in restricted sampling: the norm-preserving heuristic provides a useful reference for where to search, while empirical calibration remains important for locating a favorable operating point.

Taken together, these results establish two aspects of Sparse Neural Thickets. First, high-performing perturbations survive severe restriction of the search space, and much of the observed enrichment can already arise from reducing its dimension. Second, the choice of coordinates still modulates the high-performing tail, as seen from the weaker large-magnitude control. Low-magnitude parameters therefore provide a simple, deterministic 1% subspace with a favorable upper tail, while the norm-preserving heuristic provides a useful reference for selecting the corresponding perturbation scale. We next ask whether this restricted search space remains useful when the sampled models are selected and ensembled on held-out data.

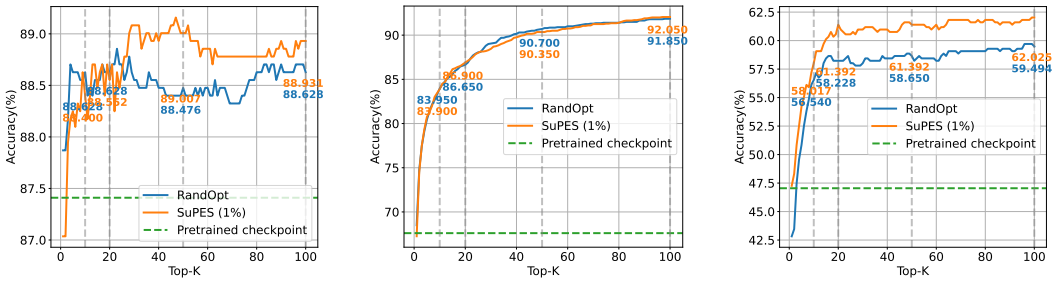
4.3 Downstream Ensemble Performance

The enrichment results above characterize the density of high-performing perturbations on the training data. We next ask whether restricting the perturbation space also preserves the downstream performance of RandOpt. We evaluate SuPES by perturbing only the lowest-magnitude 1% of the parameters and compare against full-space RandOpt.

Following the standard RandOpt setting, we use $K = 50$ as our primary ensemble size. Table 2 summarizes test accuracy across models and datasets. Despite restricting the perturbation space to only 1% of the parameters, SuPES remains competitive with full-space RandOpt across all evaluated tasks. In particular, SuPES improves over RandOpt on OlympiadBench, with an accuracy of 61.39% compared to 58.65%, and achieves 89.01% compared to 88.48% on GSM8K (77.86 versus 75.66 on Qwen2.5-1.5B-Instruct). On Countdown, the two methods attain comparable performance (90.35% versus 90.7%). These results indicate that the restricted perturbation space can retain the downstream utility of RandOpt while modifying substantially fewer parameters.

Robustness to ensemble size. A comparison at a single value of K does not reveal whether the relative performance is sensitive to the chosen ensemble size. We therefore evaluate the complete top-K trajectory for OLMo3-7B-Instruct on GSM8K, Countdown, and OlympiadBench.

For each method and dataset, we first identify an empirically favorable perturbation scale using mean reward on 200 training examples and include the immediately neighboring scales. This procedure is applied independently and identically to RandOpt and SuPES. We sample 1670 perturbations at each of three noise scales, giving a candidate pool of 5010 models per method and dataset. Candidates from all three scales are ranked jointly by training reward, and the top-K models are ensembled for test evaluation. Importantly, the test set is used only for final evaluation and does not enter either noise-scale selection or model selection. Figure 3 shows test accuracy as the ensemble size varies. The results demonstrate that the comparison at $K = 50$ is not an isolated consequence of the chosen ensemble size. On OlympiadBench, SuPES consistently outperforms RandOpt across the evaluated range of K. On GSM8K, SuPES is generally modestly better over most ensemble sizes, whereas on Countdown the two methods track each other closely and neither method uniformly dominates.



(a) GSM8K (b) Countdown (c) OlympiadBench

Figure 3: Test accuracy as a function of ensemble size K for OLMo3-7B-Instruct. For each method, perturbations are pooled across three empirically selected neighboring noise scales and ranked by training reward before constructing the top-K ensemble. SuPES (1%) improves over RandOpt or in the worst case preserve performance across ensemble sizes.

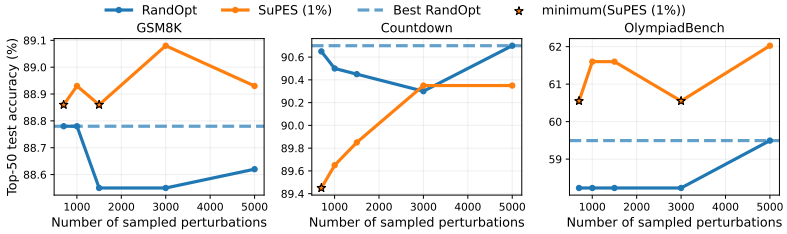


Figure 4: Top-50 ensemble test accuracy as a function of the number of sampled perturbations. SuPES frequently achieves higher accuracy at comparable sampling budgets and, in several settings, reaches the best observed RandOpt accuracy using a substantially smaller candidate pool.

Thus, the central downstream result is not that restriction invariably improves RandOpt, but rather that restricting the search to only 1% of the parameters preserves its ensemble performance and can yield systematic gains on some tasks.

These findings complement the solution-density results in Section 4.2. Sparse Neural Thickets are therefore not only visible through the density of high-performing perturbations: the restricted search space remains useful when the sampled models are selected and ensembled on held-out data.

4.4 Sampling-Budget Efficiency

The results above show that SuPES can preserve or improve downstream ensemble performance while perturbing only 1% of the model parameters. We next ask whether the restricted search space can also make better use of a limited sampling budget. To evaluate this, we construct top-50 ensembles from candidate pools of increasing size using the same noise-scale selection protocol as in Section 4.3, and compare SuPES against full-space RandOpt.

Figure 4 reports test accuracy as the number of sampled perturbations increases. Across the evaluated budgets, SuPES frequently achieves higher accuracy than RandOpt at the same number of samples. The effect is particularly clear on GSM8K and OlympiadBench, where the SuPES curve remains above the corresponding RandOpt curve throughout the evaluated range. More generally, the results show that restricting the perturbation space does not require a larger sampling budget to remain competitive with full-space sampling.

The same comparison also demonstrates that strong ensemble accuracy can often be reached with substantially fewer sampled perturbations. On GSM8K, SuPES attains 88.86% test accuracy with only 702 sampled perturbations, already exceeding the best RandOpt accuracy observed over the evaluated budgets (88.78%). Similarly, on OlympiadBench, SuPES reaches 60.55% with 702 samples, compared with a best observed RandOpt accuracy of 59.49% using up to $5{,}000$ samples. Thus,

a smaller SuPES candidate pool can already match or surpass the accuracy obtained by substantially larger full-space RandOpt pools.

Taken together, these results show two complementary benefits of sparse sampling: SuPES can improve accuracy at a comparable sampling budget, and can also reach comparable or better accuracy with fewer sampled models. This complements the solution-density results in Section 4.2: concentrating the search in a favorable sparse subspace can make the available sampling budget more effective, while still retaining the diversity benefits of sampling and ensembling.

5 DISCUSSION

Our results establish that Neural Thickets persist under severe restrictions of the search space: perturbing only 1% of the parameters can preserve both the high-performing tail and the downstream ensemble performance of full-space sampling. This suggests that useful nearby solutions are not abundant merely because modern LLMs provide an extremely high-dimensional space to search. Instead, task-improving solutions remain accessible within surprisingly sparse coordinate subspaces. Our experiments further reveal structure within these Sparse Neural Thickets. Random restriction alone can produce substantial enrichment, small magnitude subspaces often identify favorable search spaces, which aligns with findings related to parameter-efficient finetuning (Zhou et al., 2025).

The enrichment of favorable solutions opens several complementary axes for improving the efficiency of sampling-based adaptation. First, SuPES is *parameter efficient*, modifying only a small fraction of the model while remaining competitive with full-space RandOpt. Second, a higher density of useful candidates can improve *development efficiency*: fewer perturbations may need to be sampled and evaluated to construct an ensemble with comparable downstream performance. Third, richer candidate pools may improve *inference efficiency* if comparable performance can be achieved with fewer ensemble members. Our experiments provide evidence for these gains in several settings, while also showing that they are task dependent rather than universal. More broadly, Sparse Neural Thickets suggest that sampling efficiency can be improved by jointly controlling where to search, how many parameters to search, and at what scale to search.

Limitations and outlook. Our experiments do not fully disentangle dimensionality reduction from parameter selection, since random sparse subspaces can perform similarly to low-magnitude ones. Moreover, solution density is measured using training reward and therefore does not necessarily translate proportionally into held-out performance or either form of efficiency described above. We consider only coordinate subspaces and a limited set of models and tasks; other structured or task-dependent subspaces may exhibit even stronger enrichment. Finally, our results characterize local neighborhoods around pretrained models and do not address the cost of pretraining itself. Nevertheless, our results provide a concrete answer to where useful models can be found: Neural Thickets persist in remarkably sparse parameter subspaces, and their internal structure can be exploited to make sampling-based adaptation more parameter-, development-, and inference-efficient.

Conclusions We showed that Neural Thickets persist even when sampling is restricted to a small fraction of model parameters, establishing the existence of *Sparse Neural Thickets*. Low-magnitude parameter subspaces are enriched in high-performing solutions relative to large-magnitude subspaces, indicating that useful adaptation directions are not uniformly distributed around pretrained models. We further find that perturbation scale is critical for both solution density and downstream performance. Together, these findings motivate SuPES, a parameter-efficient sampling approach that achieves competitive performance while perturbing substantially fewer parameters. Understanding the mechanisms underlying this enrichment and the interaction between subspace choice and perturbation scale remains an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for claim calibration and methodological clarity checks. We have not used generative AI tools for Generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, and interpret results.

Additionally, we used generative AI tools for create or modify scientific figures, summarize or analyze existing literature, sourcing/searching for information, reviewer-style critique, writing support, argument-structure feedback, discussion of presentation choices. We have reviewed all AI-assisted work. We manually checked every entry in tables/figures generated by AI. We manually reviewed every proposed literature for potential hallucinated references. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning, December 2020. URL http://arxiv.org/abs/2012.13255. arXiv:2012.13255 [cs.LG].

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen Technical Report, September 2023. URL https://arxiv.org/abs/2309.16609v1.

Rebekka Burkholz. Convolutional and residual networks provably contain lottery tickets. In *International Conference on Machine Learning*, 2022a.

Rebekka Burkholz. Most activation functions can win the lottery without excessive depth. In *Advances in Neural Information Processing Systems*, 2022b.

Rebekka Burkholz. Batch normalization is sufficient for universal function approximation in CNNs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wOSYMHfENq.

Rebekka Burkholz, Nilanjana Laha, Rajarshi Mukherjee, and Alkis Gotovos. On the existence of universal lottery tickets. In *International Conference on Learning Representations*, 2022.

Nigel Bastian Cendra, Abdelhamid Ezzerg, Fernando Julio Cendra, Jeremias Knoblauch, and Jakob Zeitler. Can Bayesian Optimization Efficiently Find a Strong Single Expert in Neural Thickets?, August 2026. URL http://arxiv.org/abs/2608.10867. arXiv:2608.10867 [cs.LG].

Huanran Chen, Yinpeng Dong, Zeming Wei, Yao Huang, Yichi Zhang, Hang Su, and Jun Zhu. Unveiling the Basin-Like Loss Landscape in Large Language Models, March 2026. URL http://arxiv.org/abs/2505.17646. arXiv:2505.17646 [cs.LG].

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs.LG].

Arthur da Cunha, Emanuele Natale, and Laurent Viennot. Proving the lottery ticket hypothesis for convolutional neural networks. In *International Conference on Learning Representations*, 2022.

Jonas Fischer and Rebekka Burkholz. Plant 'n' seek: Can you find the winning ticket?, 2021.

Jonas Fischer, Advait Gadhikar, and Rebekka Burkholz. Towards strong pruning for lottery tickets with non-zero biases, 2021.

Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks, March 2019. URL http://arxiv.org/abs/1803.03635. arXiv:1803.03635 [cs.LG].

Advait Gadhikar, Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Sign-in to the lottery: Reparameterizing sparse training from scratch, 2025. URL https://arxiv.org/abs/2504.12801.

Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10542–10570, 2023.

Yulu Gan and Phillip Isola. Neural Thickets: Diverse Task Experts Are Dense Around Pretrained Weights, March 2026. URL http://arxiv.org/abs/2603.12228. arXiv:2603.12228 [cs.LG].

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of Search (SoS): Learning to Search in Language, April 2024. URL http://arxiv.org/abs/2404.03683. arXiv:2404.03683 [cs.LG].

Frithjof Gressmann, Zach Eaton-Rosen, and Carlo Luschi. Improving Neural Network Training in Low Dimensional Random Bases, November 2020. URL http://arxiv.org/abs/2011.04720. arXiv:2011.04720 [cs.LG].

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey, September 2024. URL http://arxiv.org/abs/2403.14608. arXiv:2403.14608 [cs.LG].

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems, June 2024. URL http://arxiv.org/abs/2402.14008. arXiv:2402.14008 [cs.CL].

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, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs.CL].

Tom Jacobs, Advait Gadhikar, Celia Rubio-Madrigal, and Rebekka Burkholz. Hyperbolic aware minimization: Implicit bias for sparsity. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=XKB5Hu0ACY.

Tom Jacobs, Rohan Jain, and Rebekka Burkholz. Horst: Composing optimizer geometries for sparse transformer training. In *Workshop on High-dimensional Learning Dynamics (HiLD) @ ICML*, 2026b. URL https://arxiv.org/abs/2605.21104.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, February 2017. URL http://arxiv.org/abs/1609.04836. arXiv:1609.04836 [cs.LG].

Taeyeong Kim, Tae Hyeon Kim, Ahhyun Kim, and Unggi Lee. Modelpinch: Diverse task experts are dense, even in 4 parameters, 06 2026.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the Intrinsic Dimension of Objective Landscapes, April 2018. URL http://arxiv.org/abs/1804.08838. arXiv:1804.08838 [cs.LG].

Qiyao Liang, Jinyeop Song, Yizhou Liu, Jeff Gore, Ila Fiete, Risto Miikkulainen, and Xin Qiu. The Blessing of Dimensionality in LLM Fine-tuning: A Variance-Curvature Perspective, January 2026. URL http://arxiv.org/abs/2602.00170. arXiv:2602.00170 [cs.LG].

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, July 2024. URL http://arxiv.org/abs/2402.09353. arXiv:2402.09353 [cs.CL].

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred Hero, and Pramod K. Varshney. A Primer on Zeroth-Order Optimization in Signal Processing and Machine Learning, June 2020. URL http://arxiv.org/abs/2006.06224. arXiv:2006.06224 [cs.LG].

Sijia Liu, Yicheng Lang, Soumyadeep Pal, Changsheng Wang, Yancheng Huang, Chongyu Fan, James Diffenderfer, Bhavya Kailkhura, and Yihua Zhang. Position: Zeroth-Order Optimization in Deep Learning Is Underexplored, Not Underpowered, May 2026. URL https://arxiv.org/abs/2605.15622v2.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-Tuning Language Models with Just Forward Passes, January 2024. URL http://arxiv.org/abs/2305.17333. arXiv:2305.17333 [cs.LG].

John X. Morris, Niloofar Mireshghallah, Mark Ibrahim, and Saeed Mahloujifar. Learning to Reason in 13 Parameters, February 2026. URL http://arxiv.org/abs/2602.04118. arXiv:2602.04118 [cs.LG].

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, April 2026. URL http://arxiv.org/abs/2512.13961. arXiv:2512.13961 [cs.CL].

Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. *Advances in Neural Information Processing Systems*, 33, 2020.

Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2599–2610, 2020.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Yinggan Xu, Roberto Dailey, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution Strategies at Scale: LLM Fine-Tuning Beyond Reinforcement Learning, February 2026. URL http://arxiv.org/abs/2509.24372. arXiv:2509.24372 [cs.LG].

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, March 2025. doi: 10.1145/3689031.3696075. URL http://arxiv.org/abs/2409.19256. arXiv:2409.19256 [cs.LG].

Weixi Song, Zuchao Li, Lefei Zhang, Hai Zhao, and Bo Du. Sparse is Enough in Fine-tuning Pre-trained Large Language Models, June 2024. URL http://arxiv.org/abs/2312.11875. arXiv:2312.11875 [cs.LG].

ModelScope Team. EvalScope: Evaluation framework for large models, 2024. URL https://github.com/modelscope/evalscope.

Rik van der Lingen. Reaction SMILES USPTO year 2023. 12 2023. doi: 10.6084/m9.figshare.24921555.v1. URL https://figshare.com/articles/dataset/Reaction_SMILES_USPTO_year_2023/24921555.

Vatsal Venkatkrishna, Nico Daheim, and Iryna Gurevych. Parameter Exploration for RLVR via Variational Learning, August 2026. URL http://arxiv.org/abs/2608.09805. arXiv:2608.09805 [cs.LG].

Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. Parameter-Efficient Fine-Tuning in Large Models: A Survey of Methodologies, April 2025. URL http://arxiv.org/abs/2410.19878. arXiv:2410.19878 [cs.CL].

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of LoRA Experts, April 2024. URL http://arxiv.org/abs/2404.13628. arXiv:2404.13628 [cs.CL].

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. QA-LoRA: Quantization-Aware Low-Rank Adaptation of Large Language Models, October 2023. URL http://arxiv.org/abs/2309.14717. arXiv:2309.14717 [cs.LG].

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting Zeroth-Order Optimization for Memory-Efficient LLM Fine-Tuning: A Benchmark, May 2024. URL http://arxiv.org/abs/2402.11592. arXiv:2402.11592 [cs.LG].

Zheyu Zhang, Shuo Yang, and Gjergji Kasneci. Consolidating Rewarded Perturbations for LLM Post-Training, May 2026. URL https://arxiv.org/abs/2605.31494v1.

Zhong Zhang, Bang Liu, and Junming Shao. Fine-tuning Happens in Tiny Subspaces: Exploring Intrinsic Task-specific Subspaces of Pre-trained Language Models, May 2023. URL http://arxiv.org/abs/2305.17446. arXiv:2305.17446 [cs.CL] version: 1.

Chao Zhou, Tom Jacobs, Advait Gadhikar, and Rebekka Burkholz. Pay Attention to Small Weights, October 2025. URL http://arxiv.org/abs/2506.21374. arXiv:2506.21374 [cs.LG].

Threshold	Low-mag. 1	Full
100%	69.2,[66.3, 72.0]	66.4 [63.4, 69.3]
105%	60 [56.9, 63.0]	58.6 [55.5, 61.6]
110%	51 [47.9, 54.1]	50.2 [47.1, 53.3]

Table A.1: Solution-density estimates for Qwen2.5-1.5B-Instruct on Countdown under empirically selected noise scales. Each estimate uses 1,000 perturbations. Values are percentages with Wilson 95% confidence intervals.

A Additional Experiments and Analysis

A.1 Solution-Density Analysis

This section provides the threshold-, perturbation-fraction-, and noise-scale-dependent results underlying the solution-density analysis in the main paper. Unless stated otherwise, each density estimate uses 1,000 independent perturbations and is reported with a 95% Wilson confidence interval. We evaluate thresholds

$$m \in \{0.90, 0.95, 0.98, 1.00, 1.03, 1.05, 1.10\} \times s(\theta)$$

and perturbation fractions $p \in \{0.01, 0.05, 0.10, 0.50, 0.75, 1.00\}$ when available.

Noise-scale protocols. Because the solution density depends strongly on the perturbation scale σ, we consider two complementary protocols. In the empirically selected protocol, σ is chosen separately for each perturbation fraction to maximize the density of high-performing perturbations at the 110% threshold. In the norm-preserving protocol, we first identify an empirically favorable scale for the 1% subspace and transfer it to larger perturbation fractions using the heuristic from Section 3.4. The latter therefore tests whether a calibrated scale can be transferred across perturbation dimensions; it does not compare independently optimized scales at every fraction, nor does it test whether the heuristic identifies the favorable 1% scale from scratch.

Additional task. Table A.1 compares the low-magnitude 1% subspace with full-space perturbation on Countdown. The estimates remain close across the three reported thresholds and their confidence intervals overlap, providing additional evidence that high-performing perturbations persist after severe restriction of the search space beyond the GSM8K setting.

Noise-scale sweeps. Figure A.1 complements the main-text scale sweep with the lowest-magnitude 10% subspace and a random 1% subspace. In both cases, the location of the favorable region depends on the perturbation dimension and subspace. The norm-preserving heuristic does not coincide exactly with the empirical optimum, but it falls in a useful range of σ and therefore provides a reference point for subsequent calibration. The remaining mismatch is consistent with the heuristic depending only on perturbation dimension and expected Euclidean norm, while the observed reward distribution also depends on the selected coordinates and the local loss landscape.

Empirically selected scales across perturbation fractions. Tables A.2 and A.3 report the full threshold sweeps for low- and large-magnitude subspaces when each perturbation fraction is evaluated near an empirically favorable scale. For the low-magnitude subspace, the high-performing tail remains present at severe restrictions, whereas lower thresholds do not exhibit the same monotone behavior. This distinction is important: enrichment of the high-performing tail does not imply that all task-preserving or lower-performing perturbations become more common. The large-magnitude sweep provides the corresponding control and shows a substantially weaker high-performing tail at the smallest perturbation fractions.

Because these fraction sweeps use 1,000 perturbations per setting, the highest-threshold estimates are based on relatively few successes. We therefore use them to characterize the qualitative trend across thresholds and fractions, while the main-text comparison based on 5,000 perturbations provides the more precise estimate at the primary 110% threshold.

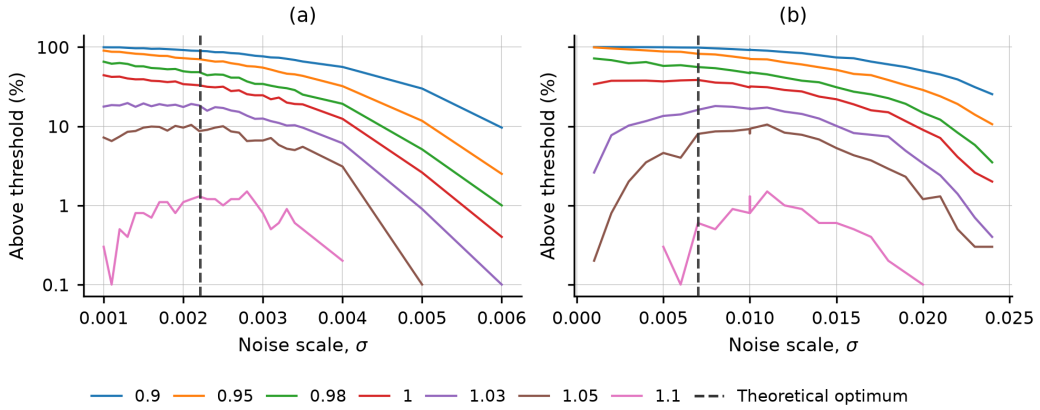


Figure A.1: Solution density as a function of the perturbation scale σ for Qwen2.5-1.5B-Instruct on GSM8K. (a) Lowest-magnitude 10% subspace. (b) Random 1% subspace. The dashed marker denotes the norm-preserving heuristic reference scale.

Threshold σ	1% 6.6×10^{-3}	5% 2.1×10^{-3}	10% 2.8×10^{-3}	50% 1.2×10^{-3}	75% 1×10^{-3}	100% 1×10^{-3}
0.90	79.0 [76.4,81.4]	94.2 [92.6,95.5]	80.3 [77.7,82.6]	90.8 [88.8,92.4]	91.5 [89.6,93.1]	91.2 [89.3,92.8]
0.95	57.3 [54.2,60.3]	75.3 [72.5,77.9]	57.5 [54.4,60.5]	69.9 [67.0,72.7]	71.5 [68.6,74.2]	70.1 [67.2,72.9]
0.98	36.6 [33.7,39.6]	49.8 [46.7,52.9]	36.8 [33.9,39.8]	46.3 [43.2,49.4]	46.7 [43.6,49.8]	44.1 [41.1,47.2]
1.00	25.1 [22.5,27.9]	33.5 [30.6,36.5]	25.4 [22.8,28.2]	31.1 [28.3,34.0]	31.4 [28.6,34.3]	30.6 [27.8,33.5]
1.03	13.1 [11.1,15.3]	16.1 [14.0,18.5]	13.6 [11.6,15.9]	14.7 [12.6,17.0]	15.2 [13.1,17.6]	15.1 [13.0,17.5]
1.05	7.3 [5.8,9.1]	8.1 [6.6,10.0]	6.5 [5.1,8.2]	6.4 [5.0,8.1]	7.1 [5.7,8.9]	7.4 [5.9,9.2]
1.10	1.5 [0.91,2.46]	1.0 [0.54,1.83]	1.5 [0.91,2.46]	0.9 [0.47,1.70]	0.8 [0.41,1.57]	0.9 [0.47,1.70]

Table A.2: Solution-density estimates across perturbation fractions for the lowest-magnitude subspace using an empirically selected noise scale for each fraction. Each scale is selected to maximize the density at the 110% threshold. Each estimate uses 1,000 perturbations; values are percentages with Wilson 95% confidence intervals.

Threshold σ	1% 8×10^{-3}	5% 8×10^{-3}	10% 4×10^{-3}	50% 2×10^{-3}	100% 1×10^{-3}
0.90	92.5 [90.7,94.0]	77.7 [75.0,80.2]	91.5 [89.6,93.1]	85.4 [83.1,87.5]	91.2 [89.3,92.8]
0.95	79.1 [76.5,81.5]	54.0 [50.9,57.1]	71.9 [69.0,74.6]	64.3 [61.3,67.2]	70.1 [67.2,72.9]
0.98	53.0 [49.9,56.1]	35.3 [32.4,38.3]	47.8 [44.7,50.9]	40.7 [37.7,43.8]	44.1 [41.1,47.2]
1.00	35.3 [32.4,38.3]	24.8 [22.2,27.6]	33.5 [30.6,36.5]	29.0 [26.3,31.9]	30.6 [27.8,33.5]
1.03	16.5 [14.3,18.9]	11.6 [9.8,13.7]	16.5 [14.3,18.9]	15.7 [13.6,18.1]	15.1 [13.0,17.5]
1.05	6.0 [4.7,7.6]	6.0 [4.7,7.6]	9.2 [7.6,11.2]	8.4 [6.8,10.3]	7.4 [5.9,9.2]
1.10	0.5 [0.21,1.17]	1.3 [0.76,2.21]	1.0 [0.54,1.83]	1.4 [0.84,2.34]	0.9 [0.47,1.70]

Table A.3: Solution-density estimates across perturbation fractions for the largest-magnitude subspace using an empirically selected noise scale for each fraction. Each scale is selected to maximize the density at the 110% threshold. Each estimate uses 1,000 perturbations; values are percentages with Wilson 95% confidence intervals.

Threshold	Low-mag. 1%	Random 1%	Full	Large-mag. 1%
100%	24.9 [22.3, 27.7]	38.4 [35.4, 41.5]	35.5 [32.6, 38.5]	36.0 [33.1, 39.0]
105%	7.3 [5.8, 9.1]	8.0 [6.5, 9.8]	5.7 [4.4, 7.3]	5.5 [4.2, 7.1]
110%	1.5 [0.91, 2.46]	0.6 [0.28, 1.30]	0.2 [0.05, 0.73]	0.1 [0.02, 0.56]

Table A.4: Solution-density estimates for Qwen2.5-1.5B-Instruct on GSM8K under the norm-preserving scale-transfer protocol. The 1% reference scale is selected empirically and the full-space scale is obtained by the heuristic transfer. Each estimate uses 1,000 perturbations. Values are percentages with Wilson 95% confidence intervals.

Norm-preserving scale transfer. Table A.4 gives the 1% coordinate-subspace comparison under the scale-transfer protocol, while Tables A.5 and A.6 extend the analysis across perturbation fractions for the low- and large-magnitude subspaces. The qualitative upper-tail pattern remains visible after transferring the calibrated 1% scale across dimensions. In particular, severe low-magnitude restriction retains a favorable high-performing tail, while the large-magnitude control remains weaker.

These results should not be read as a comparison between independently optimized sparse and full-space searches: the 1% reference scale is selected empirically and scales at larger fractions are obtained by the heuristic. Their role is instead to test whether the density pattern persists under a simple dimension-dependent transfer rule. Together with Figure A.1, they support using the heuristic as a practical reference for where to search for a suitable σ, rather than as an exact predictor of the optimum.

A.2 Fine-Grained Enrichment Across Perturbation Fractions

Figure A.2 reports pairwise enrichment ratios between perturbation fractions. The top row uses empirically selected scales, whereas the bottom row corresponds to the norm-preserving scale-transfer analysis. The left column focuses on the high-performing threshold and the right column on the task-preserving threshold.

The matrices show that enrichment is not limited to the comparison between 1% and full-space perturbation. For example, under the empirically selected scales, the 10% low-magnitude subspace is enriched relative to the 50% subspace at the high-performing threshold. In contrast, several com-

Threshold	1%	5%	10%	50%	75%	100%
0.90	77.3 [74.6,79.8]	81.9 [79.4,84.2]	89.5 [87.4,91.3]	94.9 [93.4,96.1]	91.5 [89.6,93.1]	96.7 [95.4,97.6]
0.95	54.9 [51.8,58.0]	60.1 [57.0,63.1]	70.3 [67.4,73.1]	74.8 [72.0,77.4]	71.5 [68.6,74.2]	79.7 [77.1,82.1]
0.98	36.2 [33.3,39.2]	37.5 [34.6,40.5]	48.2 [45.1,51.3]	48.6 [45.5,51.7]	46.7 [43.6,49.8]	53.1 [50.0,56.2]
1.00	24.9 [22.3,27.7]	25.3 [22.7,28.1]	32.9 [30.1,35.9]	32.3 [29.5,35.3]	31.4 [28.6,34.3]	35.5 [32.6,38.5]
1.03	12.1 [10.2,14.3]	13.1 [11.1,15.3]	18.3 [16.0,20.8]	14.5 [12.5,16.8]	15.2 [13.1,17.6]	14.9 [12.8,17.2]
1.05	7.3 [5.8,9.1]	7.1 [5.7,8.9]	8.7 [7.1,10.6]	6.8 [5.4,8.5]	7.1 [5.7,8.9]	5.7 [4.4,7.3]
1.10	1.5 [0.91,2.46]	0.6 [0.28,1.30]	1.3 [0.76,2.21]	0.5 [0.21,1.17]	0.8 [0.41,1.57]	0.2 [0.05,0.73]

Table A.5: Solution-density estimates across perturbation fractions for the lowest-magnitude subspace under the norm-preserving scale-transfer protocol. The 1% reference scale is selected empirically and scales for larger fractions are obtained by the heuristic. Each estimate uses 1,000 perturbations; values are percentages with Wilson 95% confidence intervals.

Threshold	1%	5%	10%	50%	100%
0.90	94.9 [93.4,96.1]	99.0 [98.2,99.5]	99.8 [99.3,99.9]	99.0 [98.2,99.5]	96.7 [95.4,97.6]
0.95	80.8 [78.2,83.1]	86.5 [84.2,88.5]	88.3 [86.2,90.1]	84.3 [81.9,86.4]	79.7 [77.1,82.1]
0.98	55.4 [52.3,58.5]	60.0 [56.9,63.0]	59.5 [56.4,62.5]	58.3 [55.2,61.3]	53.1 [50.0,56.2]
1.00	36.0 [33.1,39.0]	38.2 [35.2,41.3]	38.6 [35.6,41.7]	37.4 [34.5,40.4]	35.5 [32.6,38.5]
1.03	14.3 [12.3,16.6]	15.5 [13.4,17.9]	14.9 [12.8,17.2]	16.1 [14.0,18.5]	14.9 [12.8,17.2]
1.05	5.5 [4.2,7.1]	5.5 [4.2,7.1]	5.1 [3.9,6.6]	5.8 [4.5,7.4]	5.7 [4.4,7.3]
1.10	0.1 [0.02,0.56]	0.5 [0.21,1.17]	0.2 [0.05,0.73]	0.1 [0.02,0.56]	0.2 [0.05,0.73]

Table A.6: Solution-density estimates across perturbation fractions for the largest-magnitude subspace under the norm-preserving scale-transfer protocol. The 1% reference scale is selected empirically and scales for larger fractions are obtained by the heuristic. Each estimate uses 1,000 perturbations; values are percentages with Wilson 95% confidence intervals.

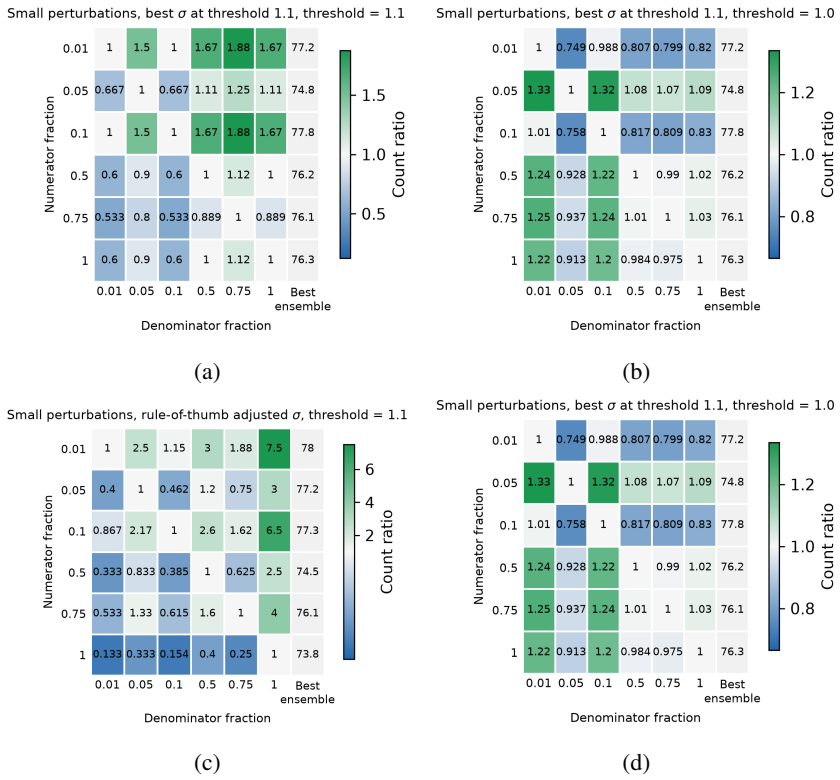


Figure A.2: Enrichment ratios between pairs of perturbation fractions for Qwen2.5-1.5B-Instruct on GSM8K. The top row uses empirically selected scales and the bottom row shows the corresponding scale-transfer analysis. Left panels use the high-performing threshold (110% of base reward); right panels use the task-preserving threshold (100%). Ratios above one indicate enrichment of the numerator fraction relative to the denominator fraction.

parisons at the task-preserving threshold show depletion under stronger restriction. This reinforces the distinction made in the main text: Sparse Neural Thickets concern the persistence and possible enrichment of the favorable upper tail, rather than a uniform increase in density at every reward level.

Under the scale-transfer protocol, the same qualitative contrast remains visible. Because the 1% scale is empirically calibrated before being transferred to larger fractions, the resulting ratios summarize how the density changes under that transfer and should not be interpreted as ratios between independently optimized operating points.

A.3 Prompts

Consistent with Team (2024); Sheng et al. (2025) and following the evaluation style of Gan & Isola (2026), we use the following task prompts.

GSM8K and OlympiadBench.

```
{question}. Let's think step by step and output the
final answer after ####.
```

Countdown.

```
Using the numbers {numbers}, create an equation
that equals {target}. You can use basic arithmetic
operations (+, -, *, /) and each number can only be
```

```
used once.  Show your work in <work> ...  </work>
tags.  Return the final answer in <answer> ...
</answer> tags, for example <answer>(1 + 2) /
3</answer>.
```

A.4 Additional Downstream and Sampling-Budget Results

Robustness to a common set of noise scales. Figure A.3 shows top-10 ensemble accuracy as σ varies for sparse and full-space perturbations. Table A.7 complements this sweep across Qwen model sizes and tasks using 1,000 sampled perturbations and the same candidate noise scales for the 4% and 100% perturbation fractions. These experiments are not the primary calibrated comparison in the main text; rather, they show that restricting the perturbation space can remain competitive even when the same coarse set of scales is used for sparse and full-space sampling.

Additional Qwen sampling-budget comparison. Table A.8 compares SuPES with 1,000 sampled perturbations against RandOpt with both 1,000 and 5,000 perturbations on Qwen-2.5-1.5B-Instruct. The results provide an additional view of sampling-budget efficiency across several ensemble sizes. Candidate models are selected by training reward only; test accuracy is evaluated after the ensemble has been formed.

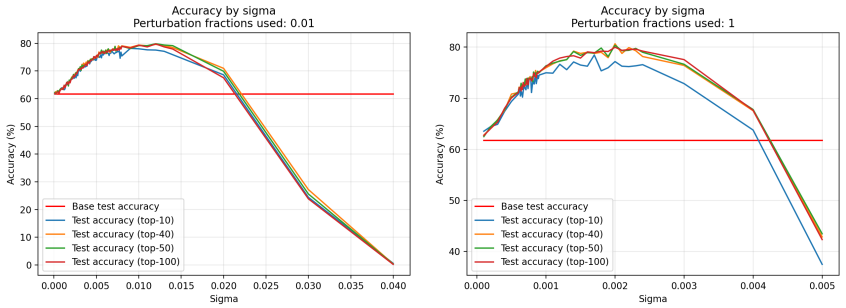
Exact values for the OLMo3-7B sampling-budget sweep. The main text reports top-50 accuracy as the number of sampled perturbations varies. Table A.9 gives the exact values underlying those curves. The comparison uses the same noise-scale selection protocol as the OLMo3-7B ensemble experiments below. Across the evaluated budgets, SuPES frequently improves the accuracy obtained from a fixed candidate pool and can reach the best observed RandOpt accuracy with a substantially smaller pool on GSM8K and OlympiadBench.

A.5 Ensemble-Size and Noise-Scale Robustness

We provide additional details for the OLMo3-7B-Instruct analysis in the main text. All scale and model selection is based only on training reward; test data are used only after the candidate models and ensemble have been selected.

Noise-scale selection. For each dataset and method, we first evaluate candidate scales on 200 training examples and select the scale with the highest mean training reward together with its two immediate neighbors. This procedure is applied independently but symmetrically to RandOpt and SuPES because restricting the perturbation dimension changes the relevant scale. We then sample 1,670 independent perturbations at each of the three selected scales, giving 5,010 candidates per method and dataset.

Tables A.10–A.12 report the training-reward density profiles at the three selected scales. They illustrate both the sensitivity of the upper tail to σ and its task dependence. In particular, the favorable



(a) (b)

Figure A.3: Top-10 test accuracy as a function of σ using 1,000 sampled perturbations. (a) Perturbing the lowest-magnitude 1% of weights. (b) Full-space perturbation.

RandOpt with 1000 samples

Dataset	Olympiadbench			GSM8K			Countdown		
perturbed weights	base	4%	100%	base	4%	100%	base	4%	100%
Qwen 0.5B	9	**12.03**	12.02	43.52	**51.18**	50.49	0	**10.9**	10.15
Qwen 1.5B	20	26.16	**28.48**	62	**76.12**	73.92	13	24.25	**30.3**
Qwen 3B	30	36.29	**37.34**	81.58	**86.43**	86.20	15.6	**46.8**	42.8

Table A.7: Top-10 test accuracy (%) from 1,000 sampled perturbations when perturbing 4% or 100% of the parameters. The same candidate noise scales are used for both perturbation fractions.

Model	Dataset	Top-K	Base	SuPES (1%), 1k	RandOpt, 1k	RandOpt, 5k
Qwen-1.5B-Instruct	GSM8K	50	62	77.47	76.35	77.41
		40	62	76.65	75.97	74.75
		10	62	76.57	74.98	73.77
Qwen-1.5B-Instruct	Countdown	50	13	38.60	39.50	38.60
		40	13	38.05	38.85	37.05
		10	13	33.75	34.00	32.10

Table A.8: Additional sampling-budget comparison on Qwen-2.5-1.5B-Instruct. These runs use a single empirically optimal noise scale selected using train reward per method.

upper tail is pronounced for SuPES on OlympiadBench, whereas the differences are smaller or more scale dependent on the other tasks. These density tables characterize the candidate pools; the downstream ensemble comparison is evaluated separately on held-out test data.

Pooling and ranking. For each method and dataset, candidates from the three selected scales are pooled and all 5,010 models are ranked jointly by training reward. For each value of K, we ensemble the top-K models using the same procedure as RandOpt.

Dependence on ensemble size. Figures A.4–A.6 show the absolute top-K ensemble accuracy together with the pointwise difference between SuPES and RandOpt. The complete trajectories show that the comparison is not determined by a single choice of K: SuPES has a persistent advantage on OlympiadBench, a smaller positive difference across much of the GSM8K range, and comparable behavior on Countdown. We use $K = 50$ as the primary operating point in the main text, while these trajectories show how the comparison evolves over a broader range of ensemble sizes.

A.6 Bootstrap Confidence Intervals

For the enrichment intervals reported in the main text, we bootstrap the observed training rewards separately within each perturbation strategy. The reward samples correspond to the empirically selected scales used in Table 1; these scales are held fixed throughout the bootstrap procedure and are not re-selected within individual bootstrap replicates. Enrichment ratios are computed relative to the full-space strategy.

For each bootstrap replicate, we draw 5,000 rewards with replacement from the 5,000 observed perturbations, recompute the solution-density estimate using Equation 2, and form the enrichment ratio using Equation 3. We repeat this procedure 10,000 times and report the 2.5th and 97.5th percentiles of the resulting enrichment distribution as the 95% bootstrap interval.

The interval therefore quantifies uncertainty under the finite perturbation-sampling protocol used in the experiment; it is not a claim of significance beyond that sampling setup..

Dataset	Method	702	1002	1500	3000	5000
GSM8K	RandOpt	88.78	88.78	88.55	88.55	88.62
	SuPES (1%)	88.86	88.93	88.86	89.08	88.93
Countdown	RandOpt	90.65	90.50	90.45	90.30	90.70
	SuPES (1%)	89.45	89.65	89.85	90.35	90.35
OlympiadBench	RandOpt	58.23	58.23	58.23	58.23	59.494
	SuPES (1%)	60.55	61.60	61.60	60.55	62.025

Table A.9: Exact top-50 test accuracies (%) underlying the sampling-budget curves in the main text for OLMo3-7B-Instruct. Columns give the number of sampled perturbations in the candidate pool.

Algorithm	Sigma	Threshold						
		0.9	**0.95**	**0.98**	**1.0**	**1.02**	**1.04**	**1.06**
RandOpt	5×10^{-4}	99.76 [99.39, 99.91]	81.92 [80.03, 83.71]	43.77 [41.44, 46.20]	19.16 [17.37, 21.13]	5.69 [4.65, 6.93]	0.90 [0.53, 1.51]	0.12 [0.03, 0.51]
RandOpt	7×10^{-4}	99.58 [99.13, 99.81]	81.44 [79.53, 83.25]	42.87 [40.54, 45.29]	18.08 [16.33, 20.01]	4.97 [4.01, 6.13]	1.14 [0.71, 1.80]	0.06 [0.00, 0.42]
RandOpt	3×10^{-4}	99.40 [98.86, 99.70]	81.02 [79.10, 82.84]	39.82 [37.52, 42.21]	15.39 [13.75, 17.19]	4.43 [3.53, 5.53]	0.72 [0.39, 1.29]	0.00 [0.00, 0.23]
SuPES (1%)	2×10^{-3}	99.52 [99.04, 99.77]	79.58 [77.60, 81.47]	39.58 [37.28, 41.97]	16.77 [15.07, 18.64]	4.13 [3.26, 5.20]	0.90 [0.53, 1.51]	0.18 [0.06, 0.59]
SuPES (1%)	3×10^{-3}	99.10 [98.54, 99.44]	76.41 [74.33, 78.43]	37.13 [34.86, 39.49]	14.55 [12.95, 16.31]	3.95 [3.10, 5.00]	0.60 [0.30, 1.14]	0.12 [0.03, 0.51]
SuPES (1%)	1×10^{-3}	99.76 [99.39, 99.91]	80.54 [78.60, 82.38]	41.98 [39.66, 44.35]	16.53 [14.84, 18.39]	3.65 [2.84, 4.66]	0.42 [0.18, 0.91]	0.06 [0.00, 0.42]

Table A.10: Training-reward solution-density estimates for OLMo3-7B-Instruct on Countdown at the three selected noise scales for RandOpt and SuPES. Each row contains 1,670 perturbations. Values are percentages with Wilson 95% confidence intervals.

Algorithm	Sigma	Threshold						
		0.9	**0.95**	**0.98**	**1.0**	**1.1**	**1.2**	**1.3**
RandOpt	7×10^{-4}	94.07 [92.83, 95.09]	82.22 [80.34, 83.99]	61.20 [58.85, 63.53]	61.20 [58.85, 63.53]	11.26 [9.84, 12.87]	0.66 [0.34, 1.22]	0.00 [0.00, 0.23]
RandOpt	5×10^{-4}	95.87 [94.81, 96.71]	83.17 [81.33, 84.89]	63.17 [60.85, 65.46]	63.17 [60.85, 65.46]	11.44 [10.01, 13.05]	0.90 [0.53, 1.51]	0.00 [0.00, 0.23]
RandOpt	1×10^{-3}	94.37 [93.16, 95.36]	82.34 [80.47, 84.11]	62.40 [60.07, 64.71]	62.40 [60.07, 64.71]	11.50 [10.07, 13.11]	1.14 [0.71, 1.80]	0.00 [0.00, 0.23]
SuPES (1%)	7×10^{-3}	93.05 [91.72, 94.16]	85.87 [84.15, 87.45]	69.82 [67.58, 72.00]	69.82 [67.58, 72.00]	19.34 [17.55, 21.32]	2.63 [1.96, 3.50]	0.24 [0.08, 0.67]
SuPES (1%)	5×10^{-3}	93.83 [92.57, 94.87]	84.25 [82.46, 85.92]	64.91 [62.61, 67.18]	64.91 [62.61, 67.18]	14.37 [12.78, 16.12]	1.56 [1.04, 2.31]	0.18 [0.06, 0.59]
SuPES (1%)	1×10^{-2}	90.42 [88.91, 91.74]	79.22 [77.21, 81.10]	63.53 [61.19, 65.80]	63.53 [61.19, 65.80]	20.78 [18.90, 22.79]	3.89 [3.06, 4.93]	0.36 [0.16, 0.78]

Table A.11: Training-reward solution-density estimates for OLMo3-7B-Instruct on OlympiadBench at the three selected noise scales for RandOpt and SuPES. Each row contains 1,670 perturbations. Values are percentages with Wilson 95% confidence intervals.

Algorithm	Sigma	Threshold						
		0.98	**0.99**	**1.0**	**1.005**	**1.01**	**1.02**	**1.03**
RandOpt	2×10^{-4}	99.82 [99.47, 99.94]	95.45 [94.34, 96.35]	87.37 [85.69, 88.88]	71.02 [68.79, 73.15]	49.10 [46.71, 51.50]	10.24 [8.88, 11.78]	0.36 [0.16, 0.78]
RandOpt	3×10^{-4}	99.52 [99.04, 99.77]	95.51 [94.40, 96.41]	87.96 [86.30, 89.43]	75.03 [72.88, 77.06]	52.40 [50.00, 54.78]	12.22 [10.72, 13.90]	0.66 [0.37, 1.18]
RandOpt	1×10^{-4}	99.52 [99.04, 99.77]	94.37 [93.16, 95.36]	85.39 [83.62, 87.00]	68.20 [65.93, 70.39]	46.53 [44.15, 48.93]	8.50 [7.26, 9.93]	0.30 [0.13, 0.70]
SuPES (1%)	1×10^{-3}	99.82 [99.47, 99.94]	96.29 [95.25, 97.10]	88.38 [86.74, 89.82]	73.95 [71.77, 76.01]	52.28 [49.88, 54.67]	13.71 [12.13, 15.46]	1.08 [0.68, 1.72]
SuPES (1%)	2×10^{-3}	99.04 [98.47, 99.40]	90.30 [88.76, 91.65]	79.70 [77.66, 81.60]	63.17 [60.83, 65.45]	41.86 [39.52, 44.24]	9.76 [8.43, 11.27]	1.26 [0.82, 1.93]
SuPES (1%)	7×10^{-4}	99.76 [99.39, 99.91]	96.29 [95.25, 97.10]	88.62 [87.01, 90.05]	73.11 [70.93, 75.18]	53.53 [51.14, 55.91]	12.87 [11.35, 14.56]	1.08 [0.68, 1.72]

Table A.12: Training-reward solution-density estimates for OLMo3-7B-Instruct on GSM8K at the three selected noise scales for RandOpt and SuPES. Each row contains 1,670 perturbations. Values are percentages with Wilson 95% confidence intervals.

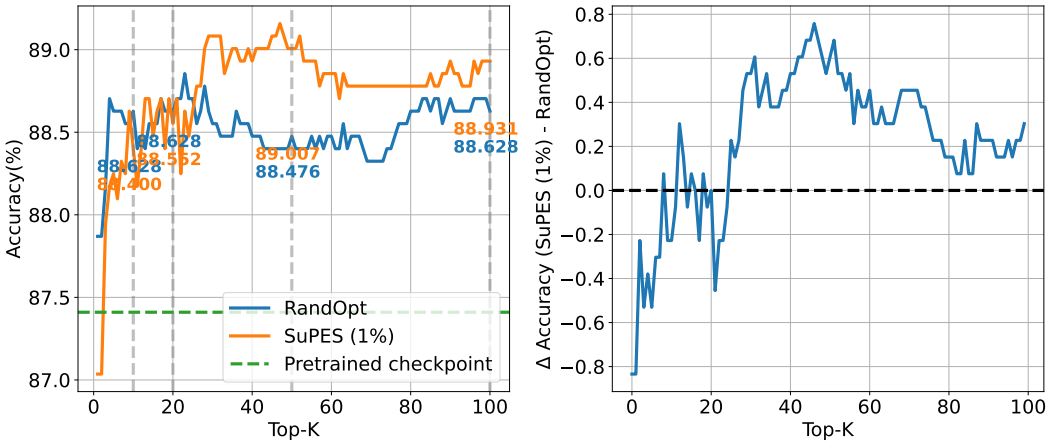


Figure A.4: OLMo3-7B-Instruct on GSM8K. Left: test accuracy of the top-K ensemble. Right: pointwise accuracy difference between SuPES and RandOpt.

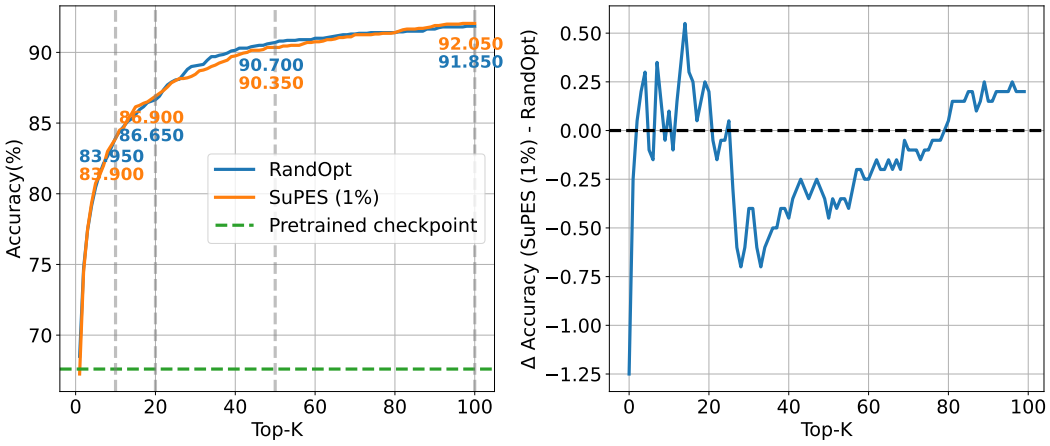


Figure A.5: OLMo3-7B-Instruct on Countdown. Left: test accuracy of the top-K ensemble. Right: pointwise accuracy difference between SuPES and RandOpt.

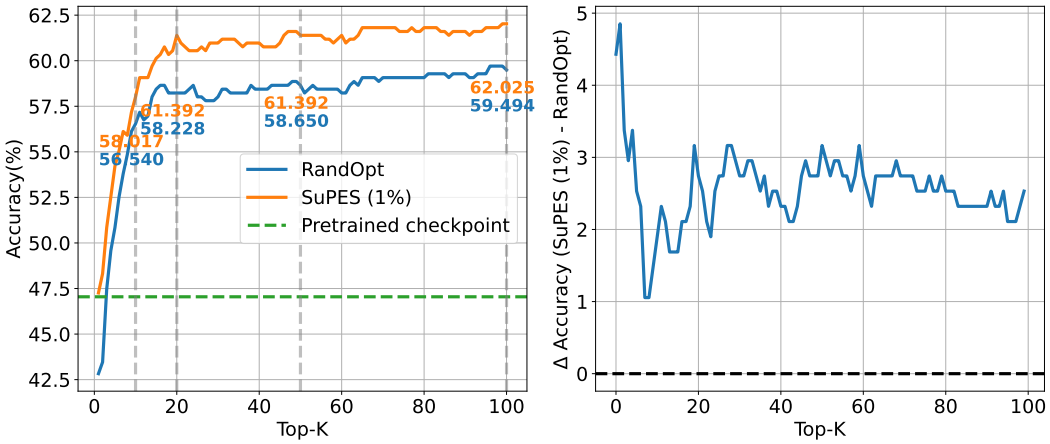


Figure A.6: OLMo3-7B-Instruct on OlympiadBench. Left: test accuracy of the top-K ensemble. Right: pointwise accuracy difference between SuPES and RandOpt.