RETHINKING NEURAL THICKETS WITHOUT WEIGHT PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The Neural Thickets hypothesis posits that pretrained language models lie in dense neighborhoods of task-specific experts in weight space, which random Gaussian perturbations can uncover. We reproduce the reported gains, but show that they do not require this geometric explanation. Perturbing a single decoder block already recovers much of the improvement, while keeping the model entirely frozen and adding a random bias to its output logits matches or exceeds full-weight perturbations on most reasoning tasks. This suggests an alternative explanation based on output diversity, validation-based selection, and voting. We formalize this view as *Perturb-Select-Vote* (PSV), a general procedure that generates perturbed predictors, selects high-performing members on a small validation set, and aggregates their predictions.

This interpretation connects Neural Thickets to established inference-time methods, including sampling, self-consistency, and verifier-guided selection. A first-order analysis further shows that small weight perturbations act primarily through their induced logit shifts. We extend this analysis to vision, where logit PSV substantially improves performance under domain shift, including on Earth observation, while full-weight perturbations provide little additional benefit. Together, these results suggest that the gains attributed to nearby weight-space experts can instead be obtained through output diversity, selection, and voting. Logit PSV realizes these benefits with a fully frozen model, providing a simple, training-free alternative to full-weight perturbation ensembles across language and vision.

1 INTRODUCTION

Gan & Isola (2026) recently reported a striking phenomenon: randomly perturbing the weights of a pretrained language model can produce many variants that outperform the original model on specific reasoning tasks. The empirical gains are substantial, reaching up to 25 accuracy points on hard reasoning benchmarks, and require little compute compared to fine-tuning. The authors propose a geometric explanation: pretrained weights of sufficiently large models are surrounded by a dense *thicket* of task-specialized models, recoverable by blind Gaussian perturbations.

This paper first asks whether the gain requires perturbing the entire network. It does not: perturbing only a small fraction of the model, or even a single decoder block, recovers much of the gain. Pushing this observation to its limit, weight perturbations can be removed altogether: the model is kept completely frozen and only its output is perturbed: each member differs by a random Gaussian bias, fixed throughout autoregressive generation, added to the logits before greedy decoding. This logit-only variant performs no additional computation inside the network and modifies no model weight. Nevertheless, it recovers, and often improves on, the gains of full-weight perturbations. On OLMo-3-7B-Instruct, it raises MATH-500 accuracy from 63.3% to 81.7%, compared with 74.7% for the full-weight perturbation ensemble, and improves USPTO from 28.6% to 44.7%.

The Neural Thickets interpretation attributes these gains to local experts in weight space. Our experiments suggest a simpler account: perturbations induce output diversity, validation-based selection retains useful members, and voting aggregates their predictions. Neural Thickets can therefore be viewed as an instance of a broader *perturb-select-vote* (PSV) template: sample random perturbations, score the resulting members on a small validation set, retain the top K, and vote at test time. This output-space view connects PSV to standard inference-time methods. A logit bias modifies the

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.

relative probability of each token, and greedy decoding under random biases defines a perturb-and-MAP sampler (Papandreou & Yuille, 2011). Selecting members by validation performance retains the biases that perform best on the target task, analogously to best-of-N and verifier-guided selection (Cobbe et al., 2021), while voting relates PSV to self-consistency (Wang et al., 2023). A first-order analysis further supports this view: small weight perturbations act, to leading order, through the logit shifts they induce. This paper then isolates the roles of sampling, selection, and voting: on strong base models, temperature-tuned sampling with majority voting already matches full-weight PSV, showing that output diversity and ensembling explain most of the gain. On weaker models, validation-based selection remains important as it filters unreliable members before aggregation.

Finally, the analysis extends to pretrained vision models under domain shift. In this setting, the analogue of a task expert is a domain expert: a perturbed representation that improves linear-probe performance on a target domain. Full-weight PSV, which perturbs the vision backbone and fits one probe per member, is compared to logit-only PSV, which fits a single probe and perturbs only the resulting class logits. The results support the same conclusion as in language: perturbing the logits is sufficient to recover the gains of full-weight perturbations, and often improves out-of-domain performance while leaving the backbone unchanged.

Our results do not rule out useful structure in the neighborhood of pretrained weights. They show, however, that such structure is not required to obtain the observed gains. PSV is better understood as output diversification, validation-based filtering, and ensembling. Beyond its role in interpreting Neural Thickets, PSV yields a simple, training-free output-space procedure that substantially improves performance on specialized tasks and domains.

Contributions.

- **Full-weight perturbations are unnecessary.** Perturbing even a single decoder block recovers much of the gain of full-weight perturbations.
- **Logit PSV.** Applying perturb-select-vote directly to the output logits of a fully frozen model matches or exceeds full-weight perturbation ensembles while modifying no weights and adding no internal computation.
- **An output-space interpretation of PSV.** Perturbations induce output diversity, selection retains high-reward members, voting aggregates their predictions, and a first-order analysis connects weight perturbations to their induced logit shifts.
- **Extension to vision.** Under domain shift, output-space perturbations recover the gains of full-weight perturbations for pretrained vision backbones as well.

2 Related Work

Test-time sampling, selection, and voting. Inference-time scaling improves fixed models by generating multiple candidates and aggregating or selecting them. Self-consistency votes over sampled reasoning paths (Wang et al., 2023), while best-of-N selects candidates using a verifier or reward model (Cobbe et al., 2021); more generally, increasing inference-time compute can substantially improve performance (Snell et al., 2025). Perturb-and-MAP provides a classical connection between perturbation and sampling, where random logit perturbations followed by MAP inference induce samples from a target distribution (Papandreou & Yuille, 2011; Maddison et al., 2017). Neural Thickets (Gan & Isola, 2026) combines these ingredients: perturbations of the weights generate candidate models, validation performance selects them, and voting aggregates their predictions.

Logit-space control. Many methods alter the behavior of a frozen model directly through its logits, including classifier-free guidance (Ho & Salimans, 2022), contrastive decoding (Li et al., 2023), post-hoc calibration (Guo et al., 2017), and watermarking, which biases a pseudo-random subset of token logits to embed a detectable signal (Kirchenbauer et al., 2023). These methods design or fit a particular logit transformation for a desired behavior. In contrast, logit PSV uses task-agnostic random biases and relies on selection and voting to exploit the resulting output diversity.

Perturbations in weight space. Previous work has revealed substantial structure in the local geometry of neural-network parameters, including mode connectivity (Garipov et al., 2018), model averaging (Izmailov et al., 2018; Wortsman et al., 2022), and low-dimensional adaptation (Li et al., 2018; Hu et al., 2022). Random weight perturbations are also central to evolution strategies and

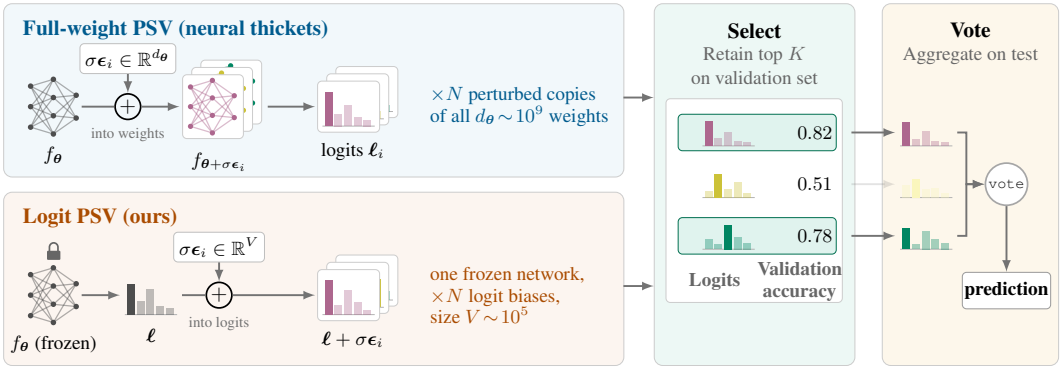


Figure 1: **Full-weight PSV versus logit PSV.** Both methods share the same selection and voting stages (*right*), but differ in the perturbation (*left*). Neural Thickets (*top*) perturb the full $d_{\boldsymbol{\theta}}$-dimensional parameter vector, producing N distinct networks. Logit PSV (*bottom*) keeps one frozen network and adds a V-dimensional bias to the output logits, fixed across decoding steps and inputs.

zeroth-order optimization (Salimans et al., 2017). Our focus is different: rather than optimizing or merging perturbations, we ask which aspects of weight-space perturbation are actually necessary for the gains of PSV. Additional related work is discussed in Appendix A.

3 The Perturb-Select-Vote Framework

We use *perturb-select-vote* (PSV) to denote a general test-time adaptation template inspired by Neural Thickets, but not restricted to full-weight perturbations. The procedure is defined by three choices: a perturbation space, a selection set, and an aggregation rule. Neural Thickets (Gan & Isola, 2026) correspond to the full-weight instance of PSV, where the perturbation space is the full parameter vector of the model, see top path of Figure 1.

Perturb-select-vote. Let $f(\cdot;\boldsymbol{\theta})$ be a pretrained predictor. PSV samples N perturbed predictors

$$f_i = \mathcal{P}\left(f(\cdot;\boldsymbol{\theta});\boldsymbol{\epsilon}_i\right), \qquad \boldsymbol{\epsilon}_i \sim \mathcal{D}_\sigma, \qquad i = 1,\ldots,N, \tag{1}$$

where $\mathcal{P}$ is a perturbation operator and $\mathcal{D}_\sigma$ is a noise distribution with scale σ. Each member is evaluated on a task-specific selection set, yielding a reward r_i. The top K members are retained:

$$\mathcal{S} = \mathrm{TopK}\left(\{r_i\}_{i=1}^{N}\right), \tag{2}$$

and the final prediction is obtained by aggregating the predictions of members $i \in \mathcal{S}$, typically by majority vote. The case $K = 1$ corresponds to selecting the single best perturbation.

Neural Thickets as full-weight PSV. RandOpt, the method introduced by Gan & Isola (2026), instantiates PSV by perturbing every model parameter. For a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$ where $d_{\boldsymbol{\theta}}$ is the number of parameters in f, each member is

$$f_i = f\left(\cdot;\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}_i\right), \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{d_{\boldsymbol{\theta}}}\right). \tag{3}$$

Each perturbed model is scored on a selection set of 200 target-task examples, and the top K models are ensembled by majority vote. Following Gan & Isola (2026), we consider perturbation budgets from $N = 1000$ to $N = 5000$ and ensemble sizes $K \in \{1, 2, 5, 10, 20, 50\}$.

Empirical effect. The original Neural Thickets study evaluates RandOpt on OLMo-3-7B-Instruct (Groeneveld et al., 2024), Qwen2.5-0.5B-Instruct, and Qwen2.5-1.5B-Instruct (Qwen Team, 2024), across seven tasks: countdown, GSM8K, MATH-500, OlympiadBench, MBPP, ROCStories, and USPTO. At $K = 50$ and $N = 5000$, the reported gains over greedy decoding reach $+25.0$ points on countdown, $+17.3$ on MATH-500, $+14.0$ on USPTO, $+11.8$ on MBPP, and $+10.3$ on OlympiadBench. The gains are smaller on tasks already well handled by the base model, such as GSM8K ($+2.3$) and ROCStories ($+0.4$). We reproduce these gains in the original experimental setting and, in some cases, improve them by tuning σ. Thus, our work does not dispute the empirical phenomenon: full-weight PSV can produce large and practically meaningful gains.

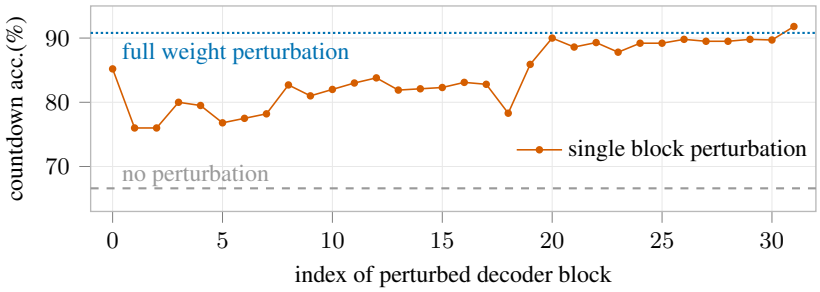


Figure 2: **Effect of perturbing individual decoder blocks.** Countdown accuracy with OLMo-3-7B-Instruct ($K = 50$), perturbing one block at a time while keeping all other weights fixed.

Weight-space interpretation. The central interpretation of Gan & Isola (2026) is that these gains reveal local task experts near the pretrained weights. This interpretation rests on three main claims. First, a *density* claim: as model size increases, a larger fraction of nearby perturbations improves task performance. Second, a *specialist* claim: perturbations that improve one task tend to hurt others, suggesting task-specific specialization. Third, a *competitiveness* claim: RandOpt with $K = 50$ mostly matches or outperforms task-adaptation methods, including reinforcement-learning methods such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), while a test-time majority-vote baseline is reported as weaker. Together, these claims frame the gains as a property of the local weight neighborhood. In the following sections, we keep the PSV procedure fixed and vary only the perturbation space, asking whether the observed gains require perturbing the full weight vector.

4 Weight Perturbations Are Not Necessary

We now investigate the mechanism underlying PSV. We progressively simplify the perturbation mechanism, first localizing it within the network and then moving it entirely to the output logits. This output-space view connects PSV to standard sampling, selection, and voting methods, allows us to isolate the source of its gains, and finally explains why internal perturbations behave similarly through the logit perturbations they induce.

4.1 Local Weight Perturbations Suffice

Full-weight PSV, as used in Neural Thickets, perturbs every parameter of the model. However, such exhaustive perturbations are not necessary. On OLMo-3-7B-Instruct for Countdown with $K = 50$, the greedy base model reaches 66.6% and full-weight PSV reaches 90.8%. As shown in Figure 2, perturbing only a single decoder block already recovers much of this gain. Every block substantially improves over the base model, with later blocks being the most effective. Thus, the gain does not require a coordinated perturbation of the full network.

The same holds within a transformer block. We perturb a single component of the first decoder block while keeping all remaining parameters frozen. Every component recovers most of the full-weight gain (Figure 3). Components that write directly into the residual stream recover the full gain, while the value and feed-forward input projections recover 96–99%. The query/key path, which acts indirectly through the attention softmax, recovers 85–89%. Qwen2.5 exhibits the same pattern from 0.5B to 3B (Appendix C). Thus, most of the PSV gain can arise from highly localized perturbations.

4.2 Perturbing Only the Logits

The preceding results raise a more direct question: are weight perturbations needed at all? We test this by moving the perturbation entirely to the output space. The model remains fully frozen, and only its output logits are perturbed during decoding. This yields a simpler PSV variant and connects PSV directly to standard sampling and ensembling methods.

Let $f(\cdot; \boldsymbol{\theta})$ be the frozen pretrained model, and let $\boldsymbol{\ell} \in \mathbb{R}^V$ denote its logits over the vocabulary V for a given token. Each PSV member i draws a vocabulary-sized Gaussian bias vector and decodes

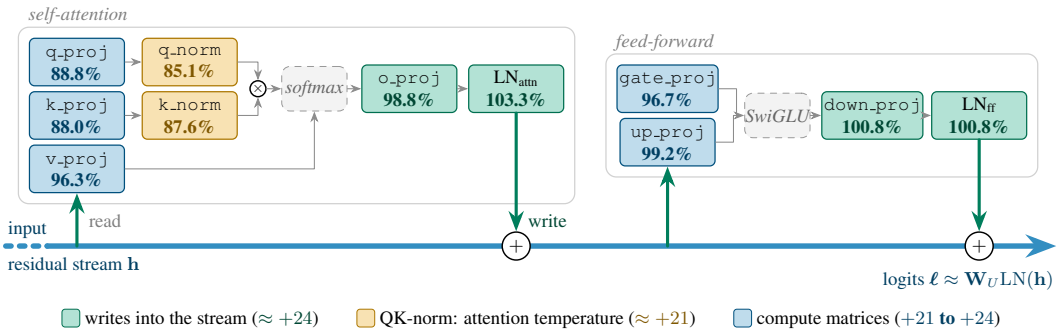


Figure 3: **Single-component perturbations within a decoder block.** Fraction of the full-weight PSV gain recovered by perturbing one component of the first decoder block of OLMo-3-7B-Instruct on Countdown with $K = 50$. All components recover at least 85% of the full-weight gain (base 66.6%, full-weight PSV 90.8%).

greedily under the perturbed logits, as shown in the bottom path of Figure 1:

$$\boldsymbol{\ell}_i = \boldsymbol{\ell} + \sigma \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_V), \tag{4}$$

where V is the vocabulary size and $\sigma > 0$ controls the noise scale. The bias $\boldsymbol{\epsilon}_i$ is fixed for member i across all decoding steps and inputs. As in full-weight PSV, we sample N members, score each one on the selection set, retain the top K, and return the majority vote at test time. This variant, which we call *logit PSV*, modifies neither the weights nor the hidden states of the model.

Despite this restriction, logit PSV matches or exceeds full-weight PSV on most reasoning tasks, as shown in Figure 4. On OLMo-3-7B-Instruct, it raises MATH-500 from 63.3% to 81.7%, compared with 74.7% for full-weight PSV at the same N and K. It also improves GSM8K from 87.5% to 91.6% and USPTO from 28.6% to 44.7%. Across the 0.5B, 1.5B, 3B, and 7B models, logit PSV closely tracks the full-weight method and often improves on it, especially on reasoning tasks (Table 5). Since this intervention performs no computation inside the network, these results show that the gains of full-weight PSV can be reproduced by injecting diversity directly in output space.

4.3 Connection to Sampling, Selection, and Voting

Logit PSV admits a simple interpretation in terms of three familiar inference-time mechanisms. We make these connections precise below and contrast them with temperature sampling and best-of-N.

A fixed logit bias induces persistent token preferences. Let x denote the input, $y_{<t}$ the sequence generated up to step t, and $p(v \mid x, y_{<t})$ the next-token distribution of the frozen model. For ensemble member i, we sample once a bias $\boldsymbol{\epsilon}_i \in \mathbb{R}^V$ and add it to the logits at every decoding step. The resulting next-token distribution is

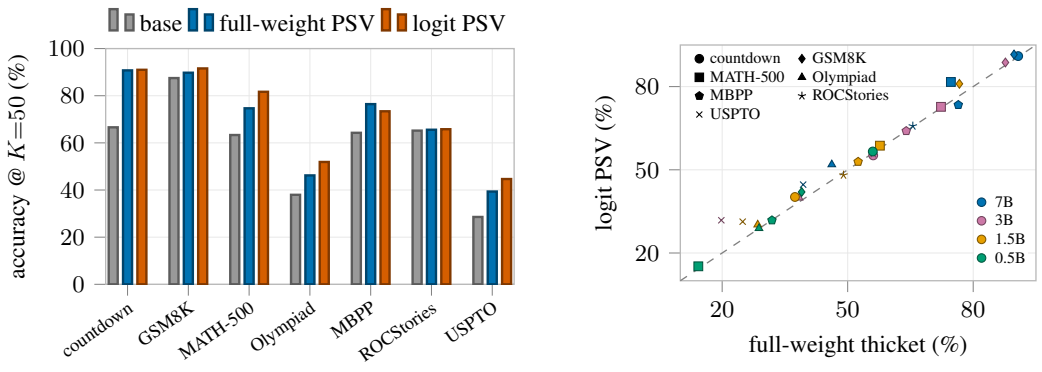
$$p_i(v \mid x, y_{<t}) \propto p(v \mid x, y_{<t}) \exp(\sigma \epsilon_{i,v}). \tag{5}$$

Thus, $\epsilon_{i,v} > 0$ systematically increases member i's relative preference for token v, while $\epsilon_{i,v} < 0$ decreases it. Sampling a bias and then greedily decoding is therefore a form of perturb-and-MAP sampling (Papandreou & Yuille, 2011): randomness is injected into the scores before taking an argmax. It differs, however, from standard temperature sampling, which can be viewed as drawing a new random perturbation at each autoregressive step. In logit PSV, the same bias $\boldsymbol{\epsilon}_i$ is reused throughout the entire decoding trajectory and across all inputs evaluated by member i. Setting $\sigma = 0$ recovers ordinary greedy decoding.

Voting corresponds to self-consistency over random perturbations. For a fixed input x, each sampled bias $\boldsymbol{\epsilon}$ defines one greedily decoded answer $a_{\boldsymbol{\epsilon}}(x)$. Since the decoding itself is deterministic once $\boldsymbol{\epsilon}$ is fixed, sampling biases induces a distribution over complete answers:

$$q_\sigma(a \mid x) = \Pr_{\boldsymbol{\epsilon}}[a_{\boldsymbol{\epsilon}}(x) = a]. \tag{6}$$

In other words, $q_\sigma(a \mid x)$ is the probability that a random logit bias leads to answer a. Before selection, different PSV members correspond to independent samples from this distribution. Majority



(a) Logit PSV matches full thicket performance **(b) Logit PSV vs. full-thicket across scales**

Figure 4: **Logit PSV matches or exceeds full-weight PSV.** *(a)* Comparative performance of the base OLMo-3-7B-Instruct model, the full-weight PSV (thicket) and the logit PSV for several tasks. Each member adds a Gaussian bias to the output logits of a frozen model. *(b)* Evaluation across scales and tasks showing that full-weights and logit PSV have similar, highly-correlated gains.

voting therefore plays the same role as self-consistency (Wang et al., 2023): it returns the answer produced most often across independently perturbed members.

Selection retains perturbations that transfer across inputs. The key difference between logit PSV and ordinary output sampling is that each sampled bias defines a persistent ensemble member. Its validation reward therefore measures how well the same token-preference shift performs across many examples from the target task. PSV ranks the sampled biases by this reward, retains the top K, and reuses those same biases on unseen test inputs.

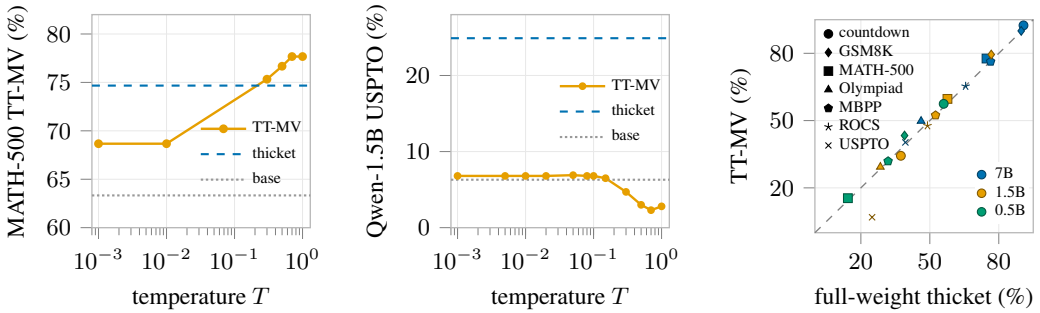
This differs from best-of-N decoding (Cobbe et al., 2021), where multiple candidate answers are generated independently for each test input and a verifier or reward model selects among them at test time. PSV instead performs selection once on a validation set and requires no test-time verifier. It also differs from temperature sampling, whose randomness is resampled independently at each decoding step and for each input; individual random draws therefore do not define reusable members whose validation performance can be transferred to future examples (see Proposition 2). Temperature-tuned majority voting (TT-MV), the test-time baseline of Gan & Isola (2026) built on self-consistency (Wang et al., 2023), shares the output diversity and voting stages of logit PSV, but omits this validation-based selection of persistent perturbations.

4.4 Disentangling the Source of PSV Gains

We next ask which parts of PSV are necessary. To isolate the effect of output diversity and voting without validation-based selection, we compare against temperature-tuned majority voting (TT-MV). This baseline majority-votes over samples of the base model at temperature T.

Test-time sampling and majority voting. Figure 5 and Table 1 show that, on strong base models, TT-MV explains much of the gain. On MATH-500 with OLMo-3-7B-Instruct, tuning the temperature raises TT-MV to 77.7%, above full-weight PSV at the matched budget (74.7%). Tuned per task, TT-MV matches or exceeds the $N = 1000$ full-weight PSV on five of seven tasks, and trails it by at most 0.3 points on the remaining two. It also exceeds the $N = 5000$ full-weight PSV on Countdown, OlympiadBench, and GSM8K, despite using a smaller search budget.

Selection matters when unfiltered samples are unreliable. TT-MV removes the validation-selection step, so its failures identify regimes where diversity and voting alone are insufficient. On weaker base models, sampling can produce diverse outputs without assigning enough probability mass to correct ones. Figure 5 shows such a case on Qwen2.5-1.5B-Instruct for USPTO: TT-MV remains at or below 6.9% for all temperatures, while full-weight PSV reaches 24.9%. This gap does not require a weight-space expert explanation. It shows that validation-based selection can be important when the base sampler is too unreliable for majority voting alone. Overall, PSV gains arise



(a) Strong-model temperature sweep

(b) Weak-model temperature sweep

(c) TT-MV vs. full-weight thicket

Figure 5: **Temperature-tuned majority voting versus full-weight PSV.** (*a*) On MATH-500 with a strong model, TT-MV matches or exceeds full-weight PSV. (*b*) On USPTO with a weaker model, TT-MV remains far below PSV, indicating that validation-based selection is useful when unfiltered samples are unreliable. (*c*) Across tasks and model scales, TT-MV closely tracks full-weight PSV, with the largest deviations occurring for weaker models and harder tasks.

Table 1: **Comparison on OLMo-3-7B-Instruct.** Accuracy at $K = 50$ for the base model, the weight average of the selected members (soup), full-weight PSV (thicket, $N = 5000$), temperature-tuned majority voting (TT-MV), and logit PSV. Output-space methods match or exceed full-weight PSV on most tasks.

task	base	soup	thicket	TT-MV	logit PSV
countdown	66.6	68.4	91.7	**92.5**	91.0
GSM8K	87.5	87.6	89.8	90.0	**91.6**
MATH-500	63.3	62.0	80.7	77.7	**81.7**
OlympiadBench	38.0	36.5	48.3	49.8	**51.9**
MBPP	64.3	66.1	76.1	**76.3**	73.4
ROCStories	65.2	65.3	65.6	65.3	**65.8**
USPTO	28.6	31.9	42.6	40.3	**44.7**

from candidate diversity, validation-based selection, and voting. Strong models may benefit from sampling and voting alone, whereas weaker models also benefit from selection.

4.5 Internal Perturbations Induce Logit Perturbations

We provide a linearized view of PSV that connects full-weight perturbations to output-space perturbations. In this view, a weight perturbation matters through its induced first-order change in the logits, and logit PSV directly injects a simpler version of the same output diversity.

Weight perturbations induce first-order logit perturbations. Let $\boldsymbol{\ell}_{\boldsymbol{\theta}}(x)$ denote the logits of a pretrained model at input x. For a perturbed model $\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}$, a first-order expansion gives

$$\boldsymbol{\ell}_{\boldsymbol{\theta}+\sigma\boldsymbol{\epsilon}}(x) \approx \boldsymbol{\ell}_{\boldsymbol{\theta}}(x) + \sigma \mathbf{J}(x)\,\boldsymbol{\epsilon}, \qquad \mathbf{J}(x) = \frac{\partial \boldsymbol{\ell}_{\boldsymbol{\theta}}(x)}{\partial \boldsymbol{\theta}}. \tag{7}$$

Thus, regardless of where the perturbation is applied in the network, its leading-order effect is a structured perturbation of the logits. This explains why perturbing even a small subset of the weights can recover most of the effect of perturbing the full network (Figure 3): any internal perturbation is propagated through the residual stream to the logits, while its location mainly determines the magnitude and structure of the resulting logit perturbation. In particular, the attention softmax attenuates perturbations along the query/key path without eliminating them (Appendix B.5). Logit PSV can therefore be viewed as a cheaper output-space surrogate: it injects diversity directly at the logits, rather than producing it indirectly through a forward pass with perturbed weights.

We empirically validate this approximation on OLMo-3-7B-Instruct for Countdown. At the perturbation scale used in our experiments, the linearized logits reproduce 89.7% of the argmax-token changes induced by the exact perturbed model; agreement is higher at smaller scales and degrades as σ increases, as expected from a first-order approximation (Table 2). This supports the interpreta-

Table 2: **Weight perturbations are well approximated by first-order logit shifts.** On OLMo-3-7B-Instruct for Countdown, we compare exact weight perturbations to their first-order logit approximation. We report the fraction of argmax-token flips induced by the exact perturbation that are reproduced by the linearized logits. The perturbation scale used in our experiments is shown in bold.

Perturbation scale σ	Agreement
2.5×10^{-4}	$97.5 \pm 1.9\%$
5×10^{-4}	$94.7 \pm 2.0\%$
$\mathbf{1 \times 10^{-3}}$	$\mathbf{89.7 \pm 2.1\%}$
2×10^{-3}	$75.9 \pm 2.6\%$
4×10^{-3}	$47.2 \pm 2.1\%$
6×10^{-3}	$25.3 \pm 1.4\%$

tion that PSV selection and voting act largely on perturbation-induced logit shifts, rather than on a qualitatively different internal computation.

5 Extending PSV to Vision

Vision differs from language in two key ways that matter for our analysis. First, LLMs reuse the same token-prediction head learned during pretraining across downstream tasks. In contrast, vision foundation models are typically adapted by fitting a new linear probe on the target-task training set. The vision probe can therefore absorb domain-specific calibration effects that remain exposed in the LLM output distribution. Second, language models generate autoregressive trajectories, so voting over sampled outputs can benefit from self-consistency (Wang et al., 2023). A vision classifier instead produces a single class prediction per image, with no analogous trajectory-level diversity.

Based on these differences, we transpose the Neural Thickets hypothesis to vision by changing the notion of specialization. We restrict our analysis to classification, but vary the image domains. We therefore define a vision *domain expert* as a perturbed backbone whose frozen representations support a better linear probe on a given target domain. If such experts exist in the local weight neighborhood, perturbing the backbone and refitting the probe should improve transfer. Conversely, if the gains arise mainly from output-space effects, perturbing only the class logits should be sufficient. We test these alternatives with a self-supervised DINOv3 backbone (Siméoni et al., 2025), comparing full-weight PSV, patch-embedding PSV, and logit PSV.

Full-weight PSV for vision. Let $f(\cdot;\boldsymbol{\theta})$ be a pretrained vision backbone. A downstream classifier is obtained by fitting a linear probe on features $f(x;\boldsymbol{\theta})$ from a labeled target-domain training set. For each full-weight PSV member, we perturb the backbone,

$$\boldsymbol{\theta}_i = \boldsymbol{\theta} + \sigma \boldsymbol{\epsilon}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_{\boldsymbol{\theta}}}), \tag{8}$$

recompute the features, and fit a new linear probe. Members are scored on a validation set, the top K are retained, and their predictions are aggregated at test time. This is our direct translation of Neural Thickets to vision: the perturbation acts on the pretrained backbone, where a neighborhood of domain-specialized representations would be expected to reside.

Patch PSV for vision. As an intermediate weight-space variant, we perturb only the patch-embedding module and refit the probe, keeping the rest of the backbone fixed. This tests whether input-level perturbations suffice for domain adaptation.

Logit PSV for vision. The output-space variant fits a single linear probe on the unperturbed frozen backbone, then keeps both backbone and probe fixed. Each PSV member is obtained by adding a Gaussian bias to the class logits, following Equation 4, with V replaced by the number of classes. The same validation-based selection and test-time voting procedure is applied. This variant changes neither the representation nor the probe and is substantially cheaper, since the probe is fitted only once rather than once per perturbation.

Closed-form recalibration. To test whether the logit-PSV gains can be explained by simple output recalibration (Saerens et al., 2002; Menon et al., 2021), we also consider a deterministic baseline. For each class c, we add the fixed bias

$$b_c = \log \pi_{\text{true}}(c) - \log \pi_{\text{pred}}(c), \tag{9}$$

Table 3: **Vision PSV under domain shift.** We report improvements relative to linear probing on the frozen DINOv3-7B backbone using worst-region accuracy for FMoW, worst-group accuracy for Waterbirds, and balanced top-1 accuracy for ImageNet. PSV variants use validation-selected ensembles with $K = 10$; closed-form denotes a single deterministic output recalibration. Weight-space perturbations yield little improvement, whereas output-space methods produce substantial gains on FMow, Waterbirds, which present strong train-to-test shift.

	DINOv3	weight-space PSV		output-space	
Dataset (metric)	base	thicket	patch-PSV	logit PSV	closed-form
ImageNet (bal. top-1)	88.4	0.0	+0.1	−0.2	n/a
FMoW-OOD (worst-region)	35.1	+1.0	+0.3	**+6.2**	+3.7
Waterbirds (worst-group)	78.9	+0.3	+0.0	**+5.0**	+5.0

where $\pi_{\mathrm{true}}(c)$ and $\pi_{\mathrm{pred}}(c)$ are respectively the true and predicted class frequencies on the validation group of interest. This correction shifts the predicted class marginal toward the observed validation marginal. For binary classification, it reduces to adjusting the decision threshold.

Results. Table 3 reports gains over the base linear probe. We consider two datasets exhibiting strong train-to-test distribution shifts for the probe: FMoW (Christie et al., 2018), evaluated with worst-region accuracy, and Waterbirds (Sagawa et al., 2020), evaluated with worst-group accuracy. We additionally include balanced ImageNet (Deng et al., 2009) as a control setting without such a shift. The two weight-space variants, full-weight and patch PSV, have little effect, improving worst-group accuracy by at most one point even for the 6.7B backbone. By contrast, logit PSV improves worst-group accuracy by 5-6 points while leaving both backbone and probe unchanged. Closed-form recalibration recovers most of this gain on FMoW; on the binary Waterbirds task, it matches logit PSV by effectively selecting an improved decision threshold. On balanced ImageNet, none of the methods provides a substantial gain.

Analysis. These results suggest that the vision gains are primarily a recalibration effect. Logit PSV improves worst-group performance by roughly 5 to 6 points on FMoW and Waterbirds, both of which exhibit substantial train-to-test shifts, but provides essentially no gain on balanced ImageNet, where train and evaluation distributions are aligned. The closed-form correction reinforces this interpretation, recovering most of the logit-PSV gain on the shifted datasets by directly recalibrating the class marginal, while again yielding no meaningful improvement on ImageNet.

Furthermore, if random backbone perturbations revealed domain-specialized representations, refitting a probe on them should improve target-domain performance. We do not observe this, even at the largest scale considered, 6.7B parameters. Instead, gains appear only when the output distribution is modified, suggesting that vision PSV is better explained by recalibration under distribution shift than by nearby domain experts in weight space.

6 Discussion

The gains of Neural Thickets do not require perturbing the full model: comparable improvements arise from perturbing a single decoder block or only the logits of a frozen model. This supports interpreting Neural Thickets through perturb-select-vote, connecting them to perturb-and-MAP sampling, self-consistency, and verifier-guided selection. In vision, gains under domain shift likewise arise primarily at the classifier output. These findings do not rule out useful local structure in weight space, but show that it is not required for the gains studied here.

Limitations. Our language experiments stop at 7B parameters; larger models may exhibit weight-space effects that logit perturbations fail to capture. PSV requires a labeled selection set and a way to compare generated answers for voting, limiting its applicability to open-ended generation. Our first-order analysis is valid only for sufficiently small σ. Finally, like full-weight PSV, logit PSV requires K model evaluations per test input, increasing inference cost relative to a single prediction.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the writing of proofs, provide feedback on research methodology and experimental design, and support method implementation. We did not use generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims or providing key ingredients of proofs, proposing hypotheses, translation, dataset cleaning or reformatting, qualitative or thematic data analysis, or interpretation of results. Additionally, we used generative AI tools for scientific figure design, suggesting experimental parameters, creating and editing software code, brainstorming, drafting and editing parts of the manuscript for clarity and readability, identifying potentially relevant literature, and proposing or refining titles and keywords.

All AI-assisted outputs were reviewed and verified by the authors: code was tested for correctness, methodological and mathematical suggestions were checked independently, literature references were verified against the original sources, and text and figures were manually revised. The authors take full responsibility for the final content of the paper, including all claims, code, figures, and text produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

All datasets and pretrained models used in this work are publicly available. The main paper and appendix document the experimental protocols, perturbation distributions, selection and voting procedures, hyperparameters, and evaluation settings used throughout the study. We will release the complete code for reproducing the LLM experiments upon publication.

ETHICS STATEMENT

This work analyzes and simplifies inference-time adaptation methods for pretrained models. All experiments use publicly available models and datasets; no new data were collected and no human subjects were involved. Logit PSV only reweights the outputs of a frozen model, so it inherits the biases and failure modes of the underlying model, and, like other training-free adaptation methods, it could in principle be used to steer a model toward undesirable behavior. We evaluate it only on standard reasoning, chemistry, and image recognition benchmarks, and we do not expect it to introduce risks beyond those of the pretrained models it builds on.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Hyeseon An, Shinwoo Park, Hyundong Jin, and Yo-Sub Han. Steering language models before they speak: Logit-level interventions. *arXiv preprint arXiv:2601.10960*, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

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? *arXiv preprint arXiv:2608.10867*, 2026.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Bryan Eikema and Wilker Aziz. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *International Conference on Computational Linguistics (COLING)*, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Yulu Gan and Phillip Isola. Neural thickets: Diverse task experts are dense around pretrained weights. In *International Conference on Machine Learning (ICML)*, 2026.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

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. In *International Conference on Learning Representations (ICLR)*, 2022.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J. Foster. Is best-of-N the best of them? coverage, scaling, and optimality in inference-time alignment. In *International Conference on Machine Learning (ICML)*, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

Mengyang Li, Fengguang Su, Ou Wu, and Ji Zhang. Class-level logit perturbation. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Muheng Li, Jian Qian, and Wenlong Mou. What should post-training optimize? A test-time scaling law perspective. *arXiv preprint arXiv:2605.10716*, 2026.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B. Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. In *Conference on Language Modeling (COLM)*, 2024.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Görür, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations (ICLR)*, 2021.

George Papandreou and Alan L. Yuille. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *International Conference on Computer Vision (ICCV)*, 2011.

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

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 2002.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Guillaume Sanchez, Alexander Spangher, Honglu Fan, Elad Levi, and Stella Biderman. Stay on topic with classifier-free guidance. In *International Conference on Machine Learning (ICML)*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.

Ved Sriraman and Adam Block. Revisiting the (sub)optimality of best-of-N for inference-time alignment. In *Conference on Learning Theory (COLT)*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zheyu Zhang, Shuo Yang, and Gjergji Kasneci. Consolidating rewarded perturbations for LLM post-training. *arXiv preprint arXiv:2605.31494*, 2026.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning (ICML)*, 2021.

Xiaoling Zhou, Wei Ye, Rui Xie, and Shikun Zhang. Mitigating spurious correlations with causal logit perturbation. *Information Sciences*, 2025.

Rethinking Neural Thickets Without Weight Perturbations

Supplementary Material

This appendix provides additional material supporting the main paper. We first present an extended discussion of related work (Section A), followed by additional theoretical analysis (Section B), cross-scale results and ablations (Section C), and further experimental details (Section D).

A Extended Related work

Test-time sampling and answer selection. A long line of work improves fixed models at inference by sampling multiple outputs and aggregating or selecting them. Self-consistency returns the majority answer over diverse reasoning paths (Wang et al., 2023); repeated sampling improves coverage as the sample count grows (Brown et al., 2024), and allocating inference compute effectively can match much larger models (Snell et al., 2025). When a verifier or reward model is available, best-of-N selects the highest-scoring candidate (Cobbe et al., 2021), while minimum Bayes risk decoding aggregates over the sampled distribution (Eikema & Aziz, 2020). Recent work further studies the coverage, optimality, and reward-model dependence of best-of-N and related test-time scaling procedures (Huang et al., 2025; Sriraman & Block, 2026; Li et al., 2026). Perturb-and-MAP provides the classical link between perturbation and sampling: adding Gumbel noise to logits and taking the arg-max yields exact categorical sampling (Papandreou & Yuille, 2011; Maddison et al., 2017; Jang et al., 2017), closely related to temperature and nucleus sampling (Holtzman et al., 2020). We place Neural Thickets (Gan & Isola, 2026) in this broader family: perturbations generate diversity, validation-based selection favors high-reward members, and voting aggregates their predictions.

Logit-space control. A growing body of work modifies model behavior directly in logit space while keeping the underlying predictor fixed. This includes watermarking, which biases selected token logits to embed a detectable signal (Kirchenbauer et al., 2023), classifier-free guidance (Ho & Salimans, 2022; Sanchez et al., 2024), contrastive decoding (Li et al., 2023), decoding-time experts (Liu et al., 2021), and proxy-tuning (Liu et al., 2024). Post-hoc calibration adjusts the outputs of a fixed model with a temperature (Guo et al., 2017) or with per-class corrections for new class priors (Saerens et al., 2002), long-tailed classes (Menon et al., 2021), or the answer biases of language models (Zhao et al., 2021). Class-level logit perturbations have also been learned for single- and multi-label classification (Li et al., 2024), sample-wise ones against spurious correlations (Zhou et al., 2025), and corpus-specific token scores steer frozen language models (An et al., 2026). These methods design or fit the logit modification for a specific task. In contrast, logit PSV samples task-agnostic random biases.

Merging and connectivity in weight space. A substantial literature studies when different solutions can be combined in weight space. Stochastic weight averaging averages checkpoints along a training trajectory to improve generalization (Izmailov et al., 2018), while model soups average independently fine-tuned models without sacrificing accuracy (Wortsman et al., 2022). Related work composes task vectors (Ilharco et al., 2023) or explicitly merges multiple specialists (Matena & Raffel, 2022; Yadav et al., 2023). These observations are closely connected to mode connectivity (Garipov et al., 2018), linear connectivity after accounting for permutation symmetries (Frankle et al., 2020; Ainsworth et al., 2023), and shared low-loss basins across tasks (Mirzadeh et al., 2021).

Low-dimensional structure and random perturbations. A substantial literature suggests that useful adaptation directions can occupy low-dimensional subspaces. Objective landscapes can have low intrinsic dimension (Li et al., 2018), language-model fine-tuning often lies in low-dimensional subspaces (Aghajanyan et al., 2021), and low-rank adapters can efficiently specialize large models (Hu et al., 2022). Relatedly, evolution strategies and zeroth-order methods optimize models by evaluating random weight perturbations (Salimans et al., 2017; Malladi et al., 2023). Unlike Neural Thickets, however, these methods explicitly optimize the perturbations rather than selecting among unstructured random draws. Concurrent work similarly explores more structured searches of the local weight neighborhood, either by consolidating rewarded perturbations (Zhang et al., 2026) or by

Algorithm 1 Logit perturb-select-vote

Require: frozen model with logits $\boldsymbol{\ell}(\cdot)$, selection reward R, noise scale σ, population size N, ensemble size K

1: **for** $i = 1, \ldots, N$ **do**
2: draw $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_V)$
3: decode greedily under logits $\boldsymbol{\ell} + \sigma\boldsymbol{\epsilon}_i$, with $\boldsymbol{\epsilon}_i$ fixed across tokens and inputs
4: $r_i \leftarrow R(i)$ reward on the validation set
5: $\mathcal{S} \leftarrow$ indices of the top K members by r_i
6: **return** the majority vote over members $i \in \mathcal{S}$ at test time

using Bayesian optimization to identify strong single experts with fewer evaluations (Cendra et al., 2026). These results are complementary to ours: structured weight-space search may be useful even if the gains of random PSV do not require a weight-space explanation.

B Persistent Perturbations, Selection, and Voting: Formal Statements

This section gives the formal statements underlying Section 4.3. Since the vote operates on final answers, we work with distributions over complete answers rather than individual tokens. Section B.1 shows that logit PSV defines persistent vocabulary biases, Section B.2 characterizes voting over the resulting answer distribution, and Section B.3 shows why selection requires perturbations that persist across inputs. Section B.4 relates logit PSV to temperature sampling; the remaining results do not depend on this connection. The complete logit PSV procedure is given in Algorithm 1.

B.1 Each member recalibrates the vocabulary

The model decodes an input x autoregressively. At step t, it emits logits $\boldsymbol{\ell}_t = \boldsymbol{\ell}(x, y_{<t}) \in \mathbb{R}^V$ given the prefix $y_{<t}$, with next-token distribution $p(\cdot \mid x, y_{<t}) = \mathrm{softmax}(\boldsymbol{\ell}_t)$. Decoding stops at an end token or after L steps, and a deterministic map ans extracts the final answer from the output sequence y. A logit-PSV member draws one bias $\sigma\boldsymbol{\epsilon} \in \mathbb{R}^V$ and decodes greedily,

$$y_t = \arg\max_v \left(\ell_{t,v} + \sigma\epsilon_v\right), \tag{10}$$

with the same $\boldsymbol{\epsilon}$ at every step and on every input. Since $\mathrm{softmax}(\boldsymbol{\ell}_t + \sigma\boldsymbol{\epsilon})_v \propto p(v \mid x, y_{<t})\, e^{\sigma\epsilon_v}$, Equation 10 greedily decodes the recalibrated model of Equation 5, which multiplies the probability of each token by a fixed factor. This is the per-token counterpart of the per-class recalibration of Section 5, with a random bias in place of a fitted one.

A member is thus a deterministic function of its bias. We write $y^{\boldsymbol{\epsilon}}(x)$ for its output and $a_{\boldsymbol{\epsilon}}(x) = \mathrm{ans}(y^{\boldsymbol{\epsilon}}(x))$ for its answer. Full-weight PSV fits the same setup, with $\boldsymbol{\epsilon}$ a weight perturbation and $y^{\boldsymbol{\epsilon}}(x)$ the greedy output of the perturbed network. In both variants, the answers of the members follow the answer distribution

$$q_\sigma(a \mid x) = \Pr_{\boldsymbol{\epsilon}}\left[a_{\boldsymbol{\epsilon}}(x) = a\right], \tag{11}$$

where the probability is over the draw of the perturbation.

B.2 Voting returns the most frequent answer

Proposition 1 (Voting over random members)**.** *Fix an input x and draw K members independently, without selection. Their answers $a_1, \ldots, a_K$ are independent, with law $q_\sigma(\cdot \mid x)$. Let $\mathcal{A}_x$ be the set of answers with positive probability, assume that the most frequent answer $a^\star = \arg\max_a q_\sigma(a \mid x)$ is unique, and let $\gamma = q_\sigma(a^\star \mid x) - \max_{a \neq a^\star} q_\sigma(a \mid x) > 0$ be its margin. The plurality vote then returns $a^\star$ with probability at least $1 - (|\mathcal{A}_x| - 1)\, e^{-K\gamma^2/2}$.*

Proof. Each answer $a_i = a_{\boldsymbol{\epsilon}_i}(x)$ is a fixed function of an independent draw $\boldsymbol{\epsilon}_i$, so the answers are independent with law $q_\sigma(\cdot \mid x)$. Outputs have at most L tokens from a finite vocabulary, so $\mathcal{A}_x$ is

finite. For $a \neq a^\star$, let $D_a = \sum_{i=1}^{K} (\mathbf{1}[a_i = a^\star] - \mathbf{1}[a_i = a])$. Its K terms are independent, take values in $[-1, 1]$, and have mean at least γ. Hoeffding's inequality gives $\Pr[D_a \leq 0] \leq e^{-K\gamma^2/2}$. The vote returns $a^\star$ whenever $D_a > 0$ for every $a \neq a^\star$, and a union bound over the $|\mathcal{A}_x| - 1$ competitors concludes. □

The same proof applies to TT-MV, with the answer distribution p_T of temperature sampling (Equation 12) in place of q_σ, which is self-consistency (Wang et al., 2023). Thus, as the number of voters grows, the unselected PSV vote converges to the most frequent answer under $q_\sigma(\cdot \mid x)$. It is therefore asymptotically correct on x when this modal answer is correct. The same interpretation applies to TT-MV with $p_T(\cdot \mid x)$.

B.3 Selection exploits persistence across inputs

A member applies the same recalibration on the validation set and on test inputs. Selection ranks members by their reward R_i on the validation set and keeps the top K. Its effect on a test input x depends on whether R_i carries information about the test answer $a_i(x)$.

Proposition 2 (Selection needs members that persist across inputs)**.** *Let the pairs $(R_i, a_i(x))$, for $i = 1, \ldots, N$, be independent and identically distributed, and let $\mathcal{S}$ contain the K members with the highest rewards, with ties broken by independent randomness. If R_i is independent of $a_i(x)$, the selected answers $(a_i(x))_{i \in \mathcal{S}}$ are independent, with the same law as $a_1(x)$. The selected vote then has the same distribution as a vote over K unselected members.*

Proof. The pairs are independent across members, and each pair has independent components. The reward vector $(R_1, \ldots, R_N)$ is therefore independent of the answer vector $(a_1(x), \ldots, a_N(x))$. The set $\mathcal{S}$ depends only on the rewards and on the tie-breaking randomness, so it is independent of the answers. Conditioned on $\mathcal{S}$, the selected answers are thus still independent draws from the law of $a_1(x)$. □

This independence condition holds if the perturbation is resampled independently for every input, as in ordinary temperature sampling. If one were to rank such samples using rewards obtained on the selection set and then resample them on a test input, the selected indices would carry no information about their test answers. Selection therefore cannot change the test-answer distribution. In PSV, by contrast, each member retains the same perturbation across the selection and test inputs. The reward $R(\boldsymbol{\epsilon})$ and the answer $a_{\boldsymbol{\epsilon}}(x)$ then depend on the same $\boldsymbol{\epsilon}$, and selection can change the answer distribution. The next statement describes this change for a large population.

Proposition 3 (Selection limit)**.** *Draw $\boldsymbol{\epsilon}^{(1)}, \ldots, \boldsymbol{\epsilon}^{(N)}$ independently from the bias prior p, and let $R(\boldsymbol{\epsilon})$ be the member validation reward, with a continuous distribution. Keep the top $K = \lceil qN \rceil$ members by reward for a fixed fraction $q \in (0, 1)$, and let τ_q be the unique level with $\Pr[R(\boldsymbol{\epsilon}) \geq \tau_q] = q$. Then as $N \to \infty$ the empirical distribution of the kept members converges weakly to p_{sel} with density*

$$p_{\mathrm{sel}}(\boldsymbol{\epsilon}) \;\propto\; p(\boldsymbol{\epsilon})\, \mathbf{1}[\, R(\boldsymbol{\epsilon}) \geq \tau_q \,] \,.$$

Proof. Let $\hat{\tau}_N$ be the K-th largest reward. The Glivenko–Cantelli theorem, the continuity of the reward law, and the uniqueness of τ_q give $\hat{\tau}_N \to \tau_q$ almost surely. For a bounded continuous function h, the kept members satisfy

$$\frac{1}{K} \sum_{i \in \mathcal{S}} h\left(\boldsymbol{\epsilon}^{(i)}\right) = \frac{N}{K} \cdot \frac{1}{N} \sum_{i=1}^{N} h\left(\boldsymbol{\epsilon}^{(i)}\right) \mathbf{1}\left[R\left(\boldsymbol{\epsilon}^{(i)}\right) \geq \hat{\tau}_N\right].$$

Replacing $\hat{\tau}_N$ by τ_q changes the right-hand average by at most $\|h\|_\infty$ times the fraction of rewards between the two levels, which vanishes almost surely by the same theorem. The strong law of large numbers and $N/K \to 1/q$ then give the limit $\mathbb{E}[h(\boldsymbol{\epsilon})\mathbf{1}[R(\boldsymbol{\epsilon}) \geq \tau_q]]/q = \mathbb{E}_{p_{\mathrm{sel}}}[h]$. □

The continuity assumption is an idealization. A reward averaged over 200 examples takes finitely many values, and ties at the threshold are then split at random. The experiments also fix $K = 50$ rather than the fraction q.

Remark 1 (When selection helps). With many selected members, the vote on x converges to the most frequent answer under the selected distribution $q_{\text{sel}}(a \mid x) = \Pr_{\boldsymbol{\epsilon} \sim p_{\text{sel}}}[a_{\boldsymbol{\epsilon}}(x) = a]$, rather than under $q_\sigma(\cdot \mid x)$. Selection can therefore help only insofar as the selection reward is informative about test behavior across members. In particular, it can change an incorrect plurality into a correct one, but can also do the reverse. This is not guaranteed: the selection set contains only 200 examples, and individual members can overfit it (Appendix C, geometry of the selected perturbations). Selection has the most room to help when the unselected answer distribution is unreliable, as in the weak-model regime of Figure 5(b).

B.4 Relation to temperature sampling

Logit PSV and temperature sampling both inject randomness at the output, but differ in how this randomness persists during decoding. Temperature-T sampling independently resamples the perturbation at every decoding step, whereas logit PSV draws a single perturbation that is reused throughout decoding and across inputs. We refer to these two cases as *resampled* and *persistent* perturbations, respectively:

$$p_T(a \mid x) = \sum_{y\,:\,\text{ans}(y)=a} \prod_t \text{softmax}\left(\boldsymbol{\ell}(x, y_{<t})/T\right)_{y_t}. \tag{12}$$

We compare the *persistent perturbation* of logit PSV, where one $\boldsymbol{\epsilon}$ is reused at every step and on every input, with *resampled perturbations*, where an independent vector $\boldsymbol{\epsilon}_t$ is drawn at every step and on every input.

B.4.1 Resampled Gumbel perturbations recover temperature sampling

Lemma 1 (Gumbel-max). *Let $g_v \sim \text{Gumbel}(0,1)$ independently, with cumulative distribution $F(x) = e^{-e^{-x}}$ and density $f(x) = e^{-x}e^{-e^{-x}}$. Then for every k,*

$$\Pr\left[\arg\max_v (\ell_v + g_v) = k\right] = \frac{e^{\ell_k}}{\sum_{j=1}^V e^{\ell_j}} = \text{softmax}(\boldsymbol{\ell})_k.$$

Proof. Condition on $g_k = z$. Token k wins when $g_v \le \ell_k - \ell_v + z$ for all $v \ne k$, an event of probability $\prod_{v\ne k} F(\ell_k - \ell_v + z)$ by independence. Integrating over z,

$$\Pr[\,y = k\,] = \int_{-\infty}^{\infty} f(z) \prod_{v\ne k} F(\ell_k - \ell_v + z)\, dz = \int_{-\infty}^{\infty} e^{-z} e^{-e^{-z}} \exp\Big(-e^{-z}\, e^{-\ell_k} \sum_{v\ne k} e^{\ell_v}\Big) dz,$$

where the product collapses because $\prod_{v\ne k} \exp(-e^{-(\ell_k - \ell_v + z)}) = \exp(-e^{-z} e^{-\ell_k} \sum_{v\ne k} e^{\ell_v})$. Combining the two exponentials and writing $Z = \sum_j e^{\ell_j}$ gives the integrand $e^{-z}\exp(-e^{-z}e^{-\ell_k}Z)$. Substitute $u = e^{-z}$, so $e^{-z}\,dz = -du$ and the limits flip to $[0,\infty)$,

$$\Pr[\,y = k\,] = \int_0^{\infty} \exp\big(-u\, e^{-\ell_k} Z\big)\, du = \frac{1}{e^{-\ell_k} Z} = \frac{e^{\ell_k}}{Z}. \qquad \square$$

Corollary 1 (The noise scale is the temperature). *For $g_v \sim \text{Gumbel}(0,1)$ and scale $\sigma > 0$, the token $\arg\max_v(\ell_v + \sigma g_v)$ has law $\text{softmax}(\boldsymbol{\ell}/\sigma)$. Hence, at a single decoding step, the bias scale σ equals the sampling temperature T.*

Proof. The arg-max is invariant to a positive rescaling of its argument, so $\arg\max_v(\ell_v + \sigma g_v) = \arg\max_v(\ell_v/\sigma + g_v)$. Apply Lemma 1 with logits $\boldsymbol{\ell}/\sigma$. $\square$

Proposition 4 (Fresh Gumbel noise samples whole sequences). *Under fresh noise with independent $\text{Gumbel}(0,1)$ entries, the decoded sequence has law $\prod_t \text{softmax}(\boldsymbol{\ell}_t/\sigma)_{y_t}$, and its answer has law $p_{T=\sigma}(\cdot \mid x)$.*

Proof. The prefix $y_{<t}$ is a function of $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_{t-1}$, which are independent of $\boldsymbol{\epsilon}_t$. Conditioned on $y_{<t}$, Corollary 1 with logits $\boldsymbol{\ell}_t$ thus gives $\Pr[y_t = k \mid y_{<t}] = \text{softmax}(\boldsymbol{\ell}_t/\sigma)_k$. The chain rule gives the law of the sequence. The answer is a deterministic function of the sequence, so its law is $p_{T=\sigma}(\cdot \mid x)$ by Equation 12. $\square$

With resampled Gumbel perturbations, voting over K decoded sequences is therefore exactly TT-MV at temperature $T = \sigma$, and selection has no effect (Proposition 2).

B.4.2 Persistent perturbations introduce memory across decisions

A persistent perturbation has the same conditional effect as a resampled perturbation whenever the next decision depends on bias coordinates that have not influenced earlier decisions. We make this precise in an idealized restricted-candidate setting. Let $C_t(h)$ be the set of candidate tokens at step t after the prefix $h = y_{<t}$, with logits set to $-\infty$ outside $C_t(h)$, and let $U(h) = \bigcup_{s<t} C_s(h_{<s})$ collect the tokens that competed at earlier steps.

Proposition 5 (Disjoint decisions match resampled perturbations)**.** *Consider a persistent perturbation with independent entries and a prefix h with $\Pr[y_{<t} = h] > 0$. If $C_t(h) \cap U(h) = \emptyset$, the next token has the same conditional law as under a resampled perturbation. With Gumbel perturbations, this law is $\mathrm{softmax}(\boldsymbol{\ell}_t/\sigma)$ restricted to $C_t(h)$. If this condition holds at every step and for every prefix, persistent and resampled perturbations induce the same distributions over sequences and answers.*

Proof. The event $\{y_{<t} = h\}$ requires that, at each step $s < t$, the token h_s wins the arg-max over $C_s(h_{<s})$. This event therefore depends only on the entries ϵ_v with $v \in U(h)$, whereas the next token depends only on those with $v \in C_t(h)$. When the two sets are disjoint, independence implies that conditioning on $\{y_{<t} = h\}$ leaves the entries in $C_t(h)$ with their prior law, exactly as under resampling. Lemma 1, applied to the logits $\boldsymbol{\ell}_t/\sigma$ restricted to $C_t(h)$, gives the Gumbel case. The final claim follows from the chain rule, as in Proposition 4. □

For the untruncated decoder used by logit PSV, $C_t(h)$ is formally the full vocabulary, so the disjointness condition does not hold beyond the first step. Nevertheless, successive decisions may effectively depend on different subsets of competitive tokens, in which case earlier choices reveal little information about the bias coordinates relevant to the next one. Proposition 5 should therefore be viewed as an exact characterization of the disjoint case, rather than as an approximation guarantee for the full decoder.

Memory. When the same tokens compete again, the prefix carries information about their entries of the bias. The next token then follows the per-step law under the posterior of $\boldsymbol{\epsilon}$ given the prefix, instead of its prior:

$$\Pr\left[y_t = k \mid y_{<t}\right] = \Pr_{\boldsymbol{\epsilon}}\left[\arg\max_v \left(\ell_{t,v} + \sigma\epsilon_v\right) = k \,\middle|\, y^{\boldsymbol{\epsilon}}_{<t} = y_{<t}\right]. \tag{13}$$

The persistent perturbation therefore couples decisions whenever the same bias coordinates matter repeatedly, both within a sequence and across inputs. Across inputs, this is precisely the persistence that makes selection informative in Section B.3.

Example: the same two tokens compete twice. Consider two tokens A and B, logits $(\delta, 0)$ at the first step and $(0, \delta)$ at the second, with $\delta > 0$, Gumbel noise, and $\sigma = 1$. The answer is the sequence itself, and the correct answer is AB. Let $p = 1/(1 + e^{-\delta}) > 1/2$. Resampled perturbations make the two choices independently, so $\Pr[AB] = p^2$, and AB is the most frequent answer. With a persistent perturbation, both choices depend on the same difference $D = g_A - g_B$, which follows a standard logistic law. The member outputs AB when $-\delta < D < \delta$, AA when $D \geq \delta$, and BB when $D \leq -\delta$, so

$$\Pr[AB] = 2p - 1, \qquad \Pr[AA] = \Pr[BB] = 1 - p, \qquad \Pr[BA] = 0. \tag{14}$$

With $\delta = 0.4$, resampled perturbations give $\Pr[AB] = 0.36$, whereas the persistent perturbation gives $\Pr[AB] = 0.20$ and $\Pr[AA] = \Pr[BB] = 0.40$, so the asymptotic vote is incorrect. The effect reverses when the correct answer repeats one choice. With logits $(\delta, 0)$ at both steps, the fixed bias gives $\Pr[AA] = p$, against p^2 under fresh noise. This example illustrates the effect of persistence: reusing the same perturbation correlates repeated decisions. Depending on the structure of the correct sequence, this correlation can either increase or decrease its probability.

B.4.3 GAUSSIAN PERTURBATIONS HAVE A LOCAL TEMPERATURE INTERPRETATION

Logit PSV uses Gaussian rather than Gumbel perturbations. The statements below characterize the resulting law of a single decision. By Proposition 5, the same law applies under a persistent perturbation whenever the relevant bias coordinates have not influenced earlier decisions.

Lemma 2 (Gaussian perturb-and-MAP limits)**.** *Let* $\epsilon_v \sim \mathcal{N}(0,1)$ *independently and* $y_\sigma = \arg\max_v(\ell_v + \sigma\epsilon_v)$*, and suppose* $\boldsymbol{\ell}$ *has a unique maximizer* $k^\star$*. Then*

$$\Pr[\, y_\sigma = k^\star \,] \xrightarrow[\sigma\to 0]{} 1, \qquad \textit{and} \qquad \Pr[\, y_\sigma = k \,] \xrightarrow[\sigma\to\infty]{} \tfrac{1}{V} \ \textit{for every } k. \tag{15}$$

Proof. For the first limit, write $\Delta_v = \ell_{k^\star} - \ell_v > 0$ for $v \neq k^\star$. A union bound gives

$$\Pr[\, y_\sigma \neq k^\star \,] \leq \sum_{v\neq k^\star} \Pr[\sigma(\epsilon_v - \epsilon_{k^\star}) > \Delta_v] = \sum_{v\neq k^\star} \overline{\Phi}\left(\frac{\Delta_v}{\sigma\sqrt{2}}\right),$$

since $\epsilon_v - \epsilon_{k^\star} \sim \mathcal{N}(0,2)$, where $\overline{\Phi}$ is the standard normal tail. Each argument diverges as $\sigma \to 0$, so each term and the finite sum vanish. For the second limit, $\arg\max_v(\ell_v + \sigma\epsilon_v) = \arg\max_v(\ell_v/\sigma + \epsilon_v)$, and $\ell_v/\sigma \to 0$ uniformly, so the maximizer converges to $\arg\max_v \epsilon_v$. The ϵ_v are independent and identically distributed with a continuous law, so this maximizer is uniform on $\{1,\dots,V\}$ by exchangeability. □

Remark 2. The exact choice probability is the multinomial probit integral

$$\Pr[\, y_\sigma = k \,] = \int_{-\infty}^{\infty} \varphi(u) \prod_{v\neq k} \Phi\left(u + \frac{\ell_k - \ell_v}{\sigma}\right) du,$$

with φ, Φ the standard normal density and distribution. It has no softmax closed form, but it is continuous in σ and connects the point mass at $k^\star$ (Lemma 2, $\sigma \to 0$) to the uniform law ($\sigma \to \infty$). The scale σ therefore controls the stochasticity of a single decision, analogously to a sampling temperature. This analogy can be made explicit for two competing tokens whose logits differ by $\Delta > 0$. Gaussian perturbations select the lower-logit token with probability $\overline{\Phi}\left(\Delta/(\sigma\sqrt{2})\right)$. Using the approximation $\Phi(z) \approx 1/\left(1 + e^{-1.702z}\right)$ gives $1/\left(1 + e^{\Delta/(0.83\sigma)}\right)$, corresponding locally to softmax sampling at temperature $T \approx 0.83\sigma$. The analogy breaks down for large logit gaps: the Gaussian probability is bounded by $\frac{1}{2}e^{-\Delta^2/(4\sigma^2)}$, whereas the corresponding softmax probability decays as $e^{-\Delta/T}$. Gaussian perturbations therefore behave similarly to temperature sampling for near-tied alternatives, while suppressing large-logit-gap alternatives more strongly.

B.5 INTERNAL PERTURBATIONS ACT AS LOGIT PERTURBATIONS

The localized perturbation results of Section 4.1 can be understood through their effect on the output logits.

Remark 3. Let $\mathbf{h} \in \mathbb{R}^d$ be the residual stream at the decoding position, read out into logits by $\boldsymbol{\ell} = \mathbf{W}_U \,\mathrm{LN}(\mathbf{h})$ with unembedding $\mathbf{W}_U \in \mathbb{R}^{V\times d}$. A write matrix $\mathbf{W}$ (an attention or feed-forward output projection) contributes additively to the stream, $\mathbf{h} = \mathbf{h}_0 + \mathbf{W}\mathbf{a}$ for the sublayer activation $\mathbf{a}$. Perturbing it by $\mathbf{W} \mapsto \mathbf{W} + \boldsymbol{\Delta}$ shifts the stream by $\boldsymbol{\Delta}\mathbf{a}$ and, to first order, the logits by

$$\delta\boldsymbol{\ell} \approx \mathbf{W}_U \mathbf{J}_{\mathrm{LN}}(\mathbf{h})\boldsymbol{\Delta}\mathbf{a}.$$

where $\mathbf{J}_{\mathrm{LN}}$ is the Jacobian of the LayerNorm. This is an additive perturbation of the output logits, as in Equation 4, although its structure is inherited from the perturbed weights and the intervening network. Logit PSV corresponds to injecting such a shift directly at the output, without propagating a perturbation through the network. The same linearization applies to other parameters within the block. Let $\mathbf{W}$ act on an input $\mathbf{x}$ inside a sublayer whose contribution to the residual stream is $g(\mathbf{W}\mathbf{x})$, where g contains the subsequent frozen operations. Perturbing $\mathbf{W} \mapsto \mathbf{W} + \boldsymbol{\Delta}$ changes this contribution to first order by $\mathbf{J}_g\boldsymbol{\Delta}\mathbf{x}$, and hence changes the logits by:

$$\delta\boldsymbol{\ell} \approx \mathbf{W}_U \mathbf{J}_{\mathrm{LN}}(\mathbf{h})\mathbf{J}_g\boldsymbol{\Delta}\mathbf{x}.$$

Unless the perturbation lies in the null space of the resulting linear map, it therefore induces a first-order change in the logits. The location of the perturbation determines the structure and magnitude

Table 4: **Accuracy of the first-order approximation.** Fraction of perturbation-induced arg-max changes reproduced by the linearized logits on OLMo-3-7B-Instruct for Countdown. Bold indicates the perturbation scale used in our experiments.

σ	Agreement
2.5×10^{-4}	$97.5 \pm 1.9\%$
5×10^{-4}	$94.7 \pm 2.0\%$
$\mathbf{1 \times 10^{-3}}$	$\mathbf{89.7 \pm 2.1\%}$
2×10^{-3}	$75.9 \pm 2.6\%$
4×10^{-3}	$47.2 \pm 2.1\%$
6×10^{-3}	$25.3 \pm 1.4\%$

of this change through the intervening Jacobians. This provides a local explanation for why perturbations applied to very different components can ultimately act through the same output space. For query and key projections, the perturbation additionally passes through the attention softmax, whose Jacobian $\mathrm{diag}\,(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$ can attenuate changes before they reach the residual stream. This is consistent with Figure 3, where perturbing the query/key path recovers 85–89% of the full-weight gain, compared with 96–103% for the other components. We interpret this ordering as evidence that the intervening computation modulates the induced logit perturbation, rather than preventing it altogether.

Together, these results describe PSV in output space: perturbations induce persistent changes in the answer distribution, selection favors perturbations that perform well on the selection set, and voting aggregates their answers. Remark 3 connects internal perturbations to this view through the logit shifts they induce.

Validating the linear approximation. Remark 3 treats the induced logit shift as locally linear in the weight perturbation. We measure the accuracy of this approximation on OLMo-3-7B-Instruct on Countdown. For a fixed teacher-forced prefix, let $\mathbf{J} = \partial \boldsymbol{\ell}_{\boldsymbol{\theta}} / \partial \boldsymbol{\theta}$ denote the logit Jacobian at the pretrained parameters. The first-order prediction for a perturbed model $\boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}$ is

$$\boldsymbol{\ell}^{\mathrm{lin}}_{\sigma,\boldsymbol{\epsilon}} = \boldsymbol{\ell}_{\boldsymbol{\theta}} + \sigma \mathbf{J}\boldsymbol{\epsilon}. \tag{16}$$

We compute $\mathbf{J}\boldsymbol{\epsilon}$ exactly with a forward-mode Jacobian–vector product and compare this prediction against the perturbed model's actual logits. Both are evaluated along fixed token sequences through teacher forcing: each prompt from the selection set followed by the base model's greedy continuation. Each model has an arg-max token at every position and we report, over the positions where the perturbation changes that token, the fraction where the linearized effect changes it to the same one. We draw 32 perturbations $\boldsymbol{\epsilon}$ and report 95% confidence intervals in Table 4.

The first-order term alone accounts for 89.7% of the greedy choices a perturbation changes. Agreement falls off as the perturbation grows (Table 4), so this describes the regime the method actually uses rather than holding everywhere. At the perturbation scale used in our experiments, the first-order approximation reproduces 89.7% of the arg-max changes induced by the perturbed model. Agreement decreases as the perturbation scale grows (Table 4), confirming that the approximation is local. These results support modeling the effect of the weight perturbation through its induced first-order logit shift in the regime used by PSV. They do not establish equivalence at the sequence level: the comparison is performed at fixed teacher-forced prefixes, whereas autoregressive decoding can amplify discrepancies across subsequent tokens.

C Cross-Scale Numbers and Ablations

Section 4.4 reports OLMo-3-7B-Instruct in the main table. The cross-scale comparisons behind Figures 4 and 5 are collected in Table 5, and the layer ablations (perturbing all but the last block, the last block only, and the logit bias) are available in the released results table. The last-block-only perturbation recovers the full gain when its noise scale is tuned per model, with the optimal scale increasing with model size. We exclude the two degenerate cells (Qwen2.5-0.5B on Countdown and USPTO, where the base model cannot perform the task) from all conclusions.

Figure 6 traces the full K-sweep behind these numbers for the full-weight thicket and the logit-bias ensemble at all three Qwen scales, where the two track each other and the gain grows with model size.

Table 5: **Cross-scale comparison of full-weight and output-space perturbations.** Accuracy at $K{=}50$ (%) for the greedy base, full-weight PSV, logit PSV, and an input-embedding-bias variant on Qwen2.5 and OLMo-3-7B-Instruct. The input-embedding variant adds a Gaussian bias to token embeddings and is evaluated on Qwen2.5 only. The noise scale σ is tuned per model. Budgets follow Section 3 ($N{=}1000$, and $N{=}2000$ for Countdown). Degenerate Qwen2.5-0.5B cells on Countdown and USPTO are omitted.

task	base	thicket	logit bias	embed bias
Qwen2.5-0.5B-Instruct				
GSM8K	44.3	56.0	**56.6**	55.2
MATH-500	29.0	39.0	**42.0**	35.7
OlympiadBench	9.5	14.3	**15.2**	13.7
MBPP	27.0	28.8	**28.9**	22.1
ROCStories	31.8	31.9	31.8	32.1
Qwen2.5-1.5B-Instruct				
countdown	12.5	37.4	**40.2**	39.1
GSM8K	60.4	76.7	**81.0**	76.6
MATH-500	48.3	57.7	**58.7**	56.0
OlympiadBench	21.5	28.5	**30.2**	28.3
MBPP	50.0	52.5	**52.9**	53.3
ROCStories	46.7	**49.0**	48.1	48.9
USPTO	6.3	24.9	**31.3**	31.9
Qwen2.5-3B-Instruct				
countdown	13.7	**56.1**	55.2	55.8
GSM8K	81.0	87.7	**88.7**	87.1
MATH-500	62.7	72.3	**72.7**	70.3
OlympiadBench	31.6	38.4	**39.9**	39.0
MBPP	59.8	64.0	64.0	63.5
ROCStories	54.5	**56.1**	55.8	55.4
USPTO	7.0	19.8	**31.8**	15.8
OLMo-3-7B-Instruct				
countdown	66.6	90.8	**91.0**	—
GSM8K	87.5	89.8	**91.6**	—
MATH-500	63.3	74.7	**81.7**	—
OlympiadBench	38.0	46.2	**51.9**	—
MBPP	64.3	**76.4**	73.4	—
ROCStories	65.2	65.6	**65.8**	—
USPTO	28.6	39.4	**44.7**	—

The mechanism holds across architectures. Figure 7 repeats the single-weight ablation of Section 4.1 on Qwen2.5-0.5B, which differs from OLMo in three ways relevant here: it has no QK-norm, it adds biases to the query, key, and value projections, and it is pre-norm rather than post-norm. The conclusion is unchanged: perturbing any single component, including the newly present QKV biases, improves over base. The ordering is also consistent with the first-order account of Appendix B.5. The weights that reach the stream through a linear or elementwise path are the most potent: at 0.5B the value and up projections ($+7.6$, $+7.1$) even exceed the output projections ($+5.5$). The weights that act only through the attention scores or through a normalized sublayer input are the weakest: the query and key projections ($+1.4$), the query and key biases ($+1.3$, $+1.1$), and the pre-norm gains ($+1.4$, $+1.7$). The pre-norms differ from the post-norms of OLMo, which scale the sublayer output written into the stream and match `o_proj`. On Countdown, the gain from every component increases with model scale (Figure 8): at 1.5B the weakest components gain $+11$ to $+16$ and the value and output projections gain $+26$ to $+27$, above the full-block perturbation; at 3B every component gains between $+27.5$ and $+39.9$. Table 6 reports the per-component gains and training-reward spreads at the 1.5B and 3B scales.

Training-reward distributions. The distribution of member rewards on the 200 selection examples, measured across all four scales (0.5B, 1.5B, 3B, 7B), shows the same picture. Every component produces a nonzero spread of member rewards, and on OLMo-3-7B-Instruct no component collapses to the base reward across all sampled members. The residual-write components (`o_proj`,

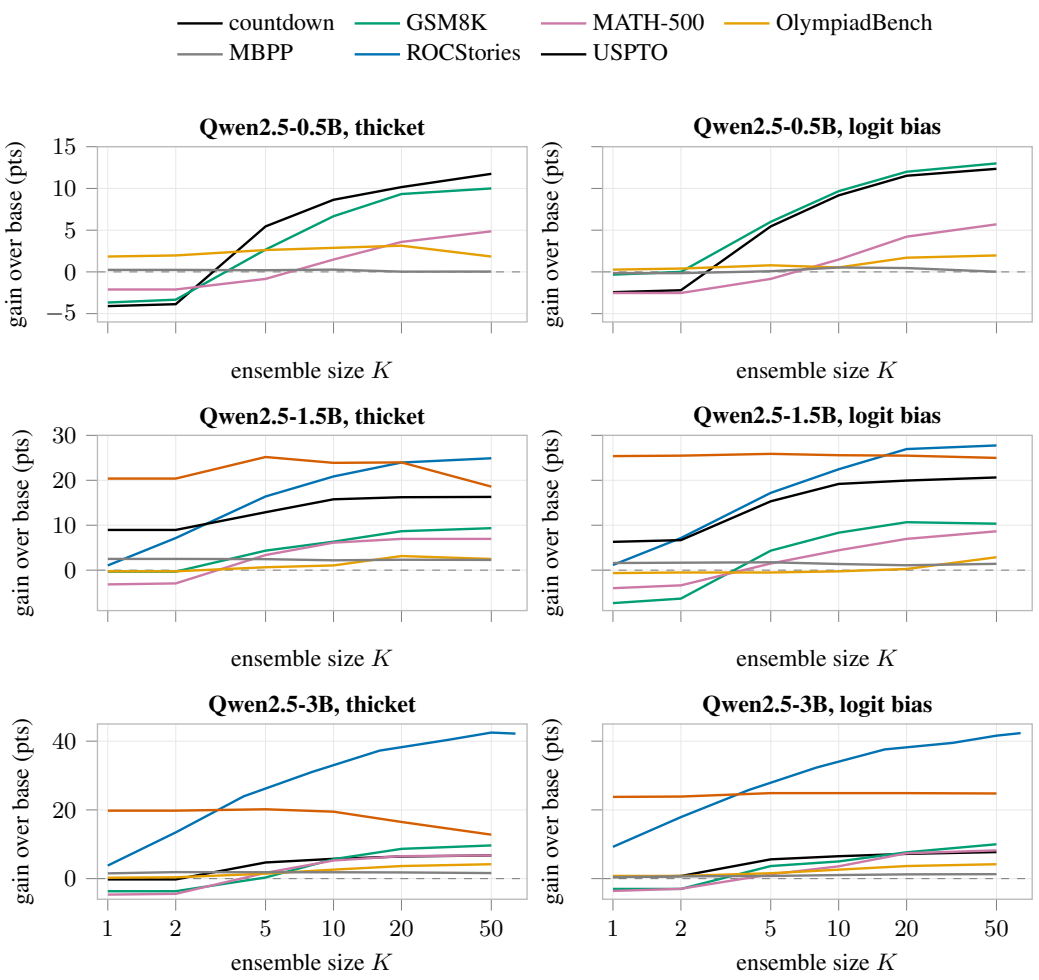


Figure 6: **Performance as a function of ensemble size.** Gain over the greedy base, in accuracy points, as a function of K for full-weight PSV and logit PSV on Qwen2.5-0.5B, 1.5B, and 3B-Instruct (N=1000, σ tuned per model). Countdown and USPTO are omitted at 0.5B because the base model cannot perform these tasks.

`down_proj`) have a standard deviation of 2 to 3 points. At 1.5B, 3B, and 7B, the compute matrices (`q/k/v_proj`, `gate/up_proj`) have a comparable spread, 1.7 to 2.5 points, with the value projection the widest at 2 to 4 points; on OLMo, 92 to 98% of their members improve over base on the selection set. The QKV biases of Qwen spread slightly less, 1.4 to 2.3 points at the same scales, and the LayerNorm gains depend on placement, as discussed above. Every component thus induces measurable variation in model behavior, providing nontrivial candidates for selection and voting.

Geometry of the selected perturbations. We test the "dense, connected neighborhood" picture directly on the OLMo-3-7B Countdown thicket. *Clustering:* the top-50 perturbation directions have a mean pairwise cosine of 6×10^{-5} (standard deviation 4.6×10^{-3}), nearly identical to random perturbation pairs (1×10^{-5}, standard deviation 4.5×10^{-3}); the selected directions show no detectable directional clustering by this statistic. *Connectivity:* averaging two top members with norm preserved gives a midpoint accuracy of 76.3% on the selection set against endpoints of 80.0%, only 2.1 points above a random-perturbation midpoint (74.2%) and well short of the endpoints, providing little evidence for a flat low-loss connection between selected members. *Generalization:* the best single member improves the selection set by 11 points but the held-out test set by only 2.4. Together with the soup collapse (Table 1), these results provide little support for interpreting the selected members as a coherent dense weight-space basin. They are instead consistent with selection over dispersed perturbations followed by output-space voting.

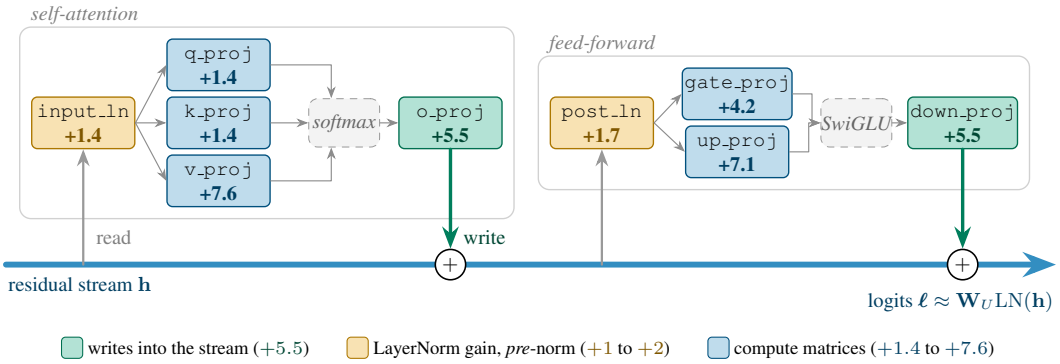
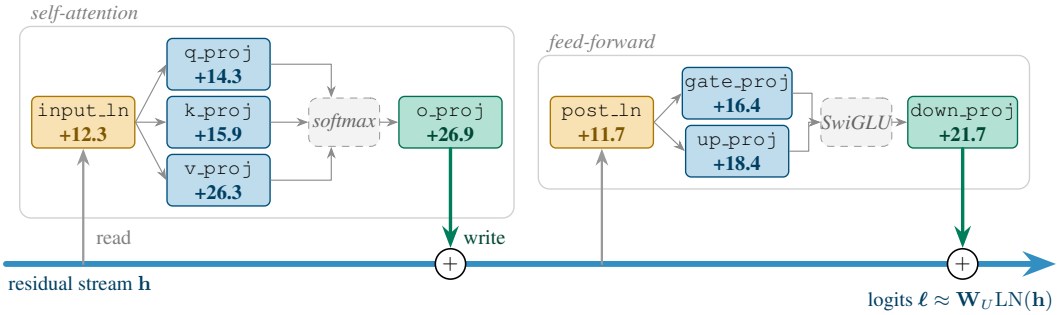


Figure 7: **Single-component perturbations transfer across architectures.** Gain over base, in accuracy points, when perturbing one component of the first decoder block of Qwen2.5-0.5B-Instruct on GSM8K (K=50; base 44.3%, full-block gain +11.8). Every component improves over base, with the largest gains from the value, up, and output projections. Qwen2.5 uses pre-norm blocks and includes QKV biases, unlike OLMo.

(a) Qwen2.5-1.5B-Instruct (countdown, base 12.5, full block +24.9)



(b) Qwen2.5-3B-Instruct (countdown, base 13.7, full block +40.3)

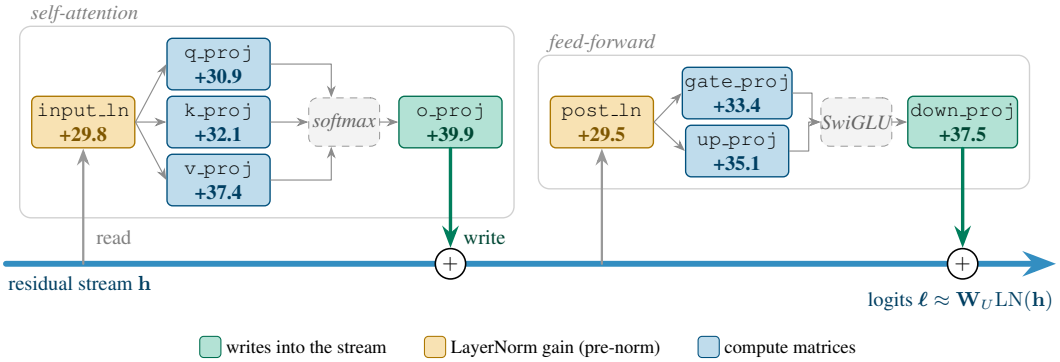


Figure 8: **Single-component perturbations remain effective at larger scales.** Gain over base, in accuracy points, when perturbing one component of the first decoder block on Qwen2.5-1.5B and 3B-Instruct for Countdown (K=32). All components improve over base, with the output and value projections among the strongest at both scales.

D Additional Experiments

D.1 The Full K-Sweep Across Methods

We report the complete ensemble-size sweep behind the output-space claims, comparing the three PSV variants head to head at every scale and task. Full-weight PSV perturbs all model weights, logit PSV adds a persistent Gaussian bias to the output logits, and TT-MV performs temperature-tuned sampling followed by plurality voting, without reward-based selection. Each is swept over $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ on four model scales (0.5B, 1.5B, 3B, 7B) and three tasks (Countdown,

Table 6: **Component-wise gains and selection-reward variability across scales.** For Qwen2.5-1.5B and 3B-Instruct on Countdown (K=32), we perturb one component of the first decoder block and report its gain over base and the standard deviation of member rewards on the selection set. Every component produces nonzero reward variability and improves the final vote, with the largest effects generally obtained from the value and output projections.

	1.5B		3B	
single weight perturbed	gain	reward std	gain	reward std
residual writes				
`self_attn.o_proj`	+26.9	2.6	+39.9	3.0
`mlp.down_proj`	+21.7	2.2	+37.5	2.4
LayerNorm gains (pre-norm)				
`input_layernorm`	+12.3	1.4	+29.8	2.2
`post_attention_layernorm`	+11.7	1.4	+29.5	2.2
compute matrices				
`self_attn.q_proj`	+14.3	1.7	+30.9	2.3
`self_attn.k_proj`	+15.9	1.7	+32.1	2.3
`self_attn.v_proj`	+26.3	3.6	+37.4	4.2
`mlp.gate_proj`	+16.4	1.8	+33.4	2.4
`mlp.up_proj`	+18.4	2.0	+35.1	2.5
biases				
`self_attn.q_proj.bias`	+11.0	1.4	+28.1	2.2
`self_attn.k_proj.bias`	+10.9	1.4	+27.5	2.2
`self_attn.v_proj.bias`	+12.4	1.5	+30.7	2.3

GSM8K, MATH-500). Figure 9 shows raw majority-vote accuracy against the greedy base, and Figure 10 the same runs as gain over base. Across these sweeps, accuracy generally increases with K, with larger gains at larger model scales. Logit PSV closely tracks or exceeds full-weight PSV on the reasoning tasks, while TT-MV falls behind primarily in the weak-model regime where reward-based selection is most useful (e.g. 0.5B and 1.5B Countdown).

D.2 Are Weight and Logit Perturbations Complementary?

If weight and logit perturbations provided complementary sources of useful variation, combining them could improve over either intervention alone. We test this directly by composing the two perturbations. For each model and task, we summarize each method by its peak accuracy over the evaluated K values and compare four ensembles: full-weight PSV alone, logit PSV alone, full-weight PSV applied on top of logit PSV (*weight* $\circ$ *logit*), and, as a control, a second logit perturbation composed with the first using a smaller noise range (*logit* $\circ$ *logit*). The control tests whether any benefit from combining weight and logit perturbations exceeds that obtained by simply composing two output-space perturbations.

Figure 11 reports the corresponding peak gains over base, and Figure 12 compares the combined method with the better single intervention and with the logit-on-logit control. On ten of twelve model–task pairs, the combined method matches or trails the better single intervention, and it shows no systematic advantage over the logit-on-logit control. Only 7B GSM8K and 7B MATH-500 improve over both comparisons, and in both cases the gain remains below one point. We therefore find little evidence that weight and logit perturbations provide complementary gains, consistent with the view that they largely act through the same output-space mechanism.

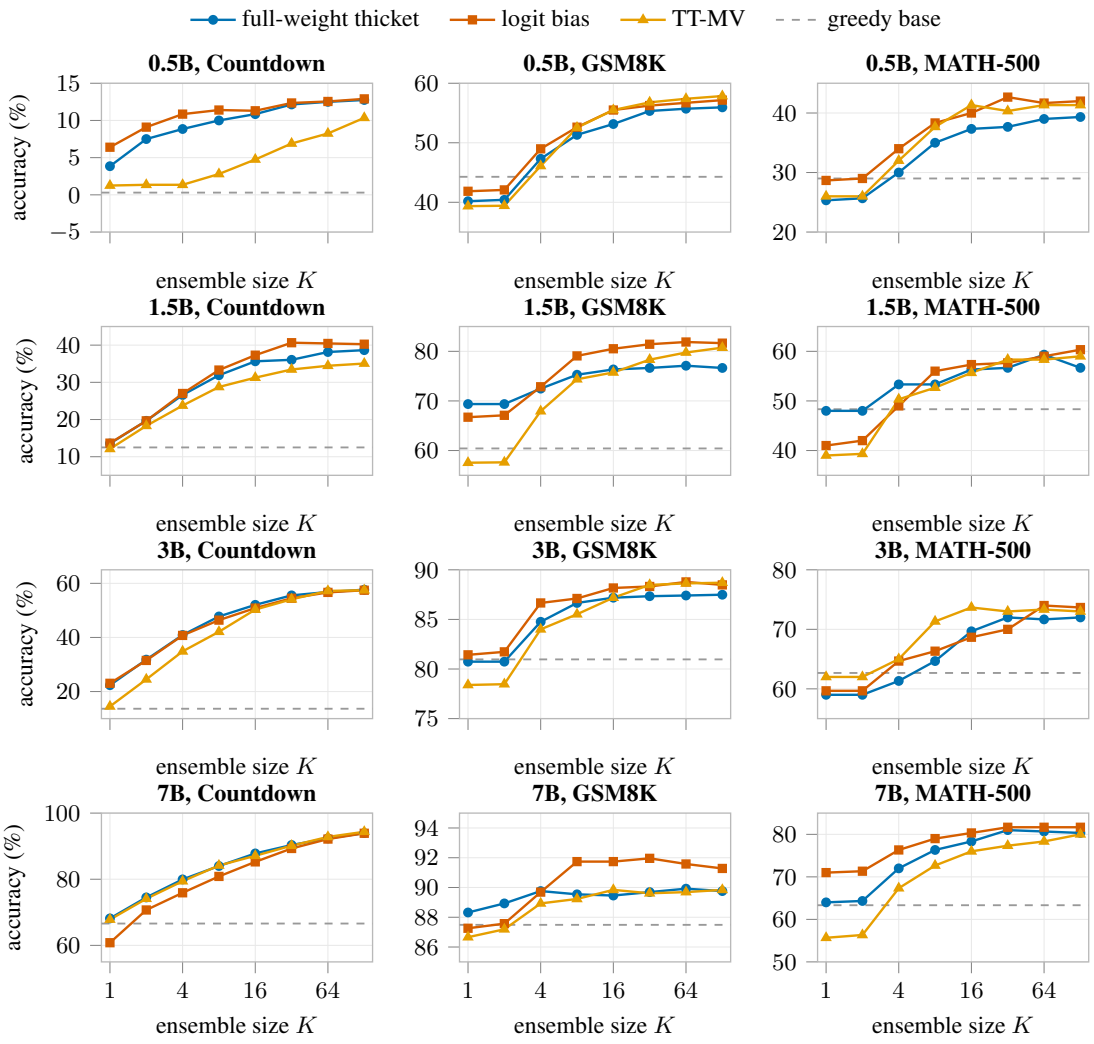


Figure 9: **Accuracy as a function of ensemble size K.** Plurality-vote accuracy for full-weight PSV, logit PSV, and TT-MV across model scales (rows) and tasks (columns). The dashed line indicates the greedy-base accuracy.

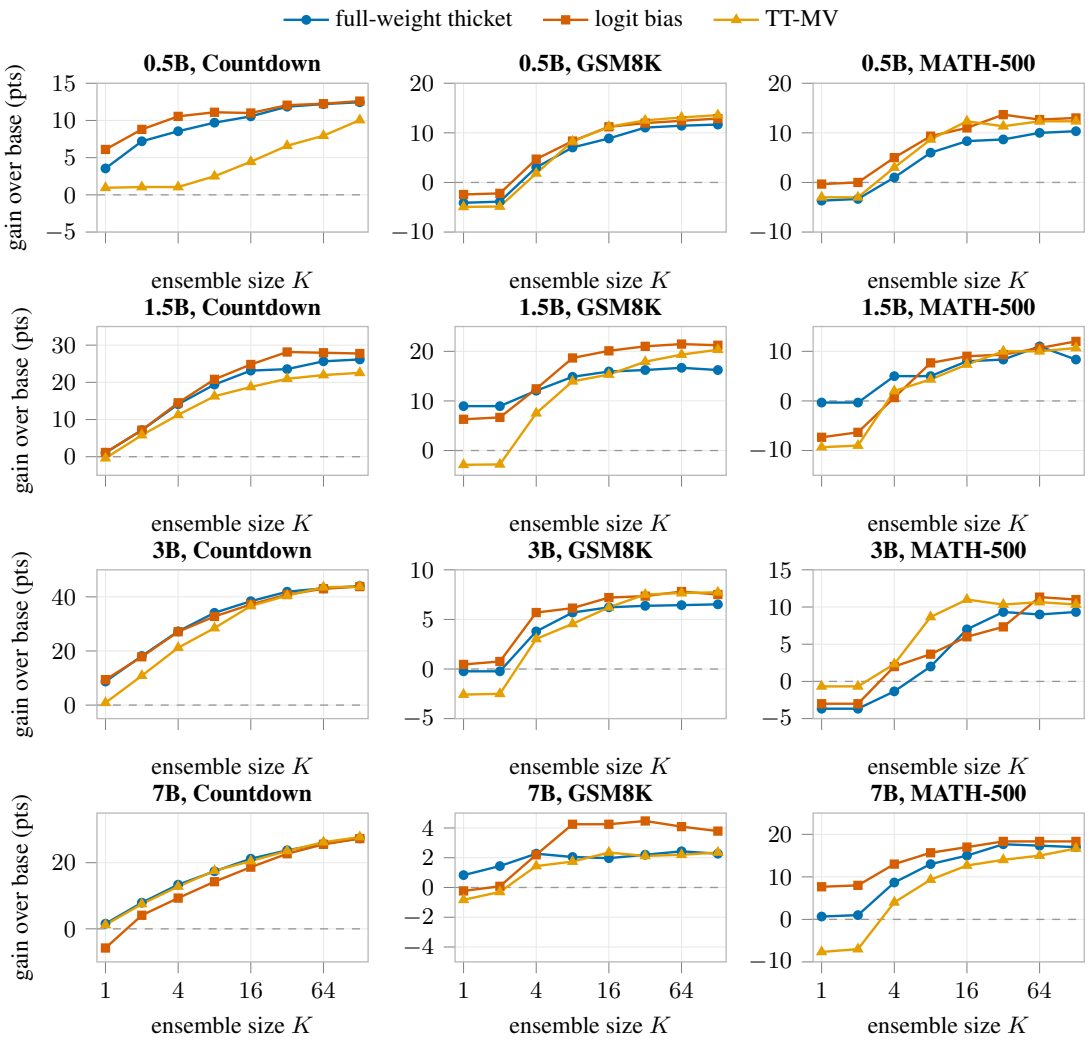


Figure 10: **Gain over the greedy base as a function of ensemble size K.** Same runs as Figure 9, shown as the accuracy gain Δ over the greedy base, across model scales (rows) and tasks (columns). The dashed line marks zero gain.

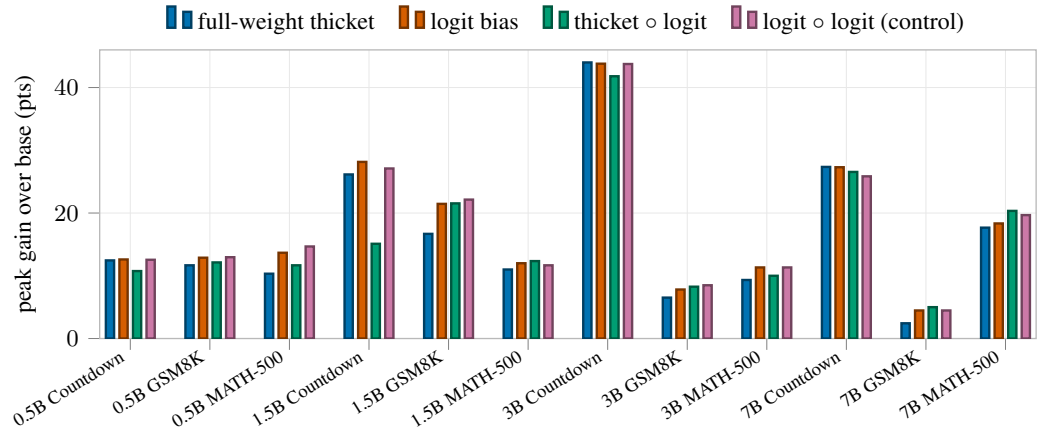


Figure 11: **Accuracy gains from individual and stacked perturbation mechanisms.** Peak gain over greedy decoding across ensemble sizes K, comparing full-weight PSV, logit PSV, their combination, and a same-mechanism control that stacks two logit biases with a smaller noise range. The symbol "∘" denotes successive application of two perturbations. Combining weight and logit perturbations rarely improves upon the stronger individual method or the logit-only control, except on two 7B reasoning tasks.

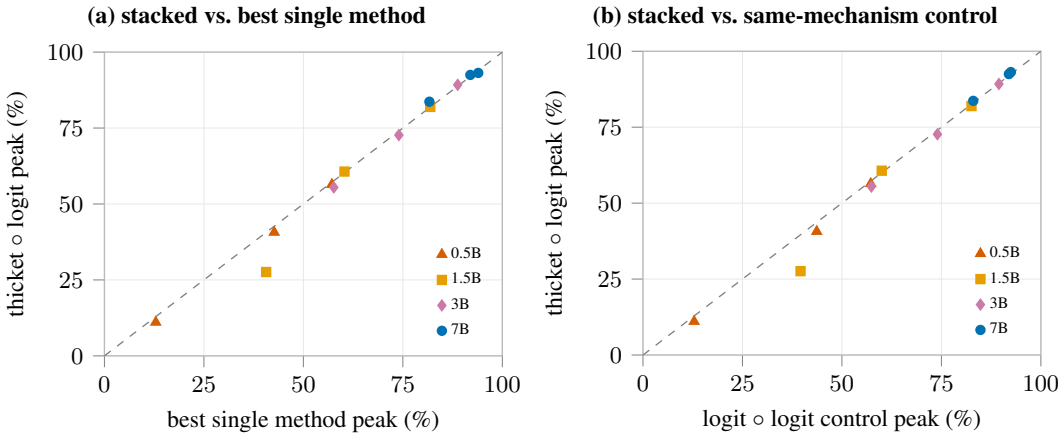


Figure 12: **Peak accuracy of stacked perturbations versus individual methods and a logit-only control.** Full-weight PSV stacked on logit PSV is compared against (a) the stronger individual method and (b) the same-mechanism control, which stacks two logit biases. The dashed diagonal indicates equal accuracy. Most points lie on or below it, with gains of approximately one point limited to two 7B reasoning tasks.