BACK TO THE WEIGHTS: TRAINING-FREE MODEL ENHANCEMENT VIA TRAINING TRAJECTORY EXTRAPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuned checkpoints need not be locally optimal in weight space. We study the endpoint displacement $\mathbf{\Delta} = \theta_{\text{end}} - \theta_{\text{start}}$ between compatible start and end checkpoints and ask whether extrapolating beyond θ_{end} can improve downstream performance without gradient computation. Cosine learning-rate decay motivates this question when the checkpoint lineage and schedule are known, but our method itself requires only the two checkpoints. We present TrajectoryForge, a gradient-free framework that extrapolates along $\mathbf{\Delta}$ with *per-matrix* stochastic scaling factors and selects a single candidate, incurring no deployment-time inference overhead. Across 7 model pairs (1.5B–14B parameters), the best sampled candidate matches or exceeds the endpoint for 6 of 7 models, with up to 7.1% relative improvement on HumanEval; two pairs also improve at the mean level. Finite-batch Hessian diagnostics identify atypical geometry along $\mathbf{\Delta}$, and a matched control yields roughly $2\times$ the success rate of isotropic search. Cross-benchmark evaluations further provide evidence that selected improvements can transfer beyond the selection benchmark.

1 INTRODUCTION

The standard paradigm for adapting large language models (LLMs) to downstream tasks involves fine-tuning on task-specific data (Ouyang et al., 2022; Wei et al., 2022), after which the resulting checkpoint is deployed as-is. This workflow implicitly treats the training endpoint θ_{end} as the best model obtainable from the training process—an assumption that, as we show, is systematically violated.

A growing line of work demonstrates that the weight space around a fine-tuned model contains exploitable structure. Model Soups (Wortsman et al., 2022) averages multiple checkpoints; Task Arithmetic (Ilharco et al., 2023) and ExPO (Zheng et al., 2024) edit models via task vectors with a single global scaling factor; RandOpt (Gan & Isola, 2026) shows that isotropic random perturbations around the *pretrained* model can match fine-tuning when combined with top-K ensemble voting. Despite their promise, none of these methods systematically leverage the *training trajectory itself* as a privileged search direction—they either require expensive multi-run training, use a single global scaling factor, or ignore directional information altogether.

Our work starts from the observation that an endpoint can remain locally improvable along the displacement $\mathbf{\Delta} = \theta_{\text{end}} - \theta_{\text{start}}$. When a known training run uses cosine learning-rate decay, its late-stage step sizes provide one motivation for testing beyond the endpoint. More generally, $\mathbf{\Delta}$ is an observable checkpoint displacement that may retain task-relevant structure even when the optimizer, schedule, and intermediate trajectory of a released model are unavailable.

We introduce TrajectoryForge[1], a gradient-free framework that extrapolates along $\mathbf{\Delta}$ with *per-matrix* stochastic scaling factors. Given only two checkpoints (θ_{start} and θ_{end}), TrajectoryForge constructs and evaluates candidate models, then deploys the best-performing candidate as a single model—requiring **zero gradient computation** and incurring **no deployment-time inference overhead**. A

[1]Code will be released upon acceptance.

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.

global alpha scan along $\mathbf{\Delta}$ (Section 5, Fig. 2) reveals an inverted-U performance curve peaking beyond the endpoint; per-matrix search further amplifies this gain.

Our main contributions are:

- We propose TrajectoryForge, a gradient-free post-hoc method that extrapolates an endpoint displacement with per-matrix stochastic scaling. Across 7 model pairs (1.5B–14B), the best sampled candidate matches or exceeds the endpoint for 6 of 7 models (up to 7.1% relative improvement on HumanEval), while two models show statistically significant mean-level improvement ($p < 10^{-20}$).
- We provide finite-batch Hessian spectral diagnostics showing that the endpoint displacement can occupy atypical geometry—high directional curvature with low projection onto the top eigenvectors—offering a possible mechanism for the roughly $2\times$ success-rate advantage over isotropic search in our matched control.
- Cross-task evaluation provides evidence for a **direction-task alignment** principle: directed perturbation improves IFEval by 1.1 percentage points while producing no practically meaningful change on MMLU, and cross-benchmark selection over 529 candidates transfers between MBPP and HumanEval.

2 Related Work

Post-hoc Weight Optimization. Model Soups (Wortsman et al., 2022) averages weights from multiple fine-tuned models; Task Arithmetic (Ilharco et al., 2023) edits models via task vectors $\tau = \theta_{\text{ft}} - \theta_{\text{pre}}$ with a single global α; subsequent extensions include TIES-Merging (Yadav et al., 2023), DARE (Yu et al., 2024), Model Stock (Jang et al., 2024), Twin-Merging (Lu et al., 2024), and Evolutionary Model Merge (Akiba et al., 2025). These methods all require *multiple* independently trained models and use global (or per-task) scaling that cannot capture per-layer variation. Our method extracts directional information from a *single* run's two checkpoints with per-matrix stochastic scaling.

Weight-Space Extrapolation. Closest to our work is ExPO (Zheng et al., 2024), which extrapolates from an SFT checkpoint toward an aligned checkpoint via a single global scalar: $\theta_{\text{ExPO}} = \theta_{\text{aligned}} + \alpha(\theta_{\text{aligned}} - \theta_{\text{SFT}})$. Our global alpha scan (Fig. 2) instantiates this same functional form and thus serves as an ExPO / Task Arithmetic baseline (+3.1 percentage points on Qwen-Coder-1.5B). TrajectoryForge generalizes along three axes: (i) per-matrix scaling with $K \sim 291$ independent scalars unlocking layer-wise heterogeneity, reaching 7.1% relative improvement on the same model; (ii) applicability to generic base→instruct SFT rather than SFT→aligned RLHF pairs; (iii) Hessian spectral grounding identifying the "sharp valley" geometry making $\mathbf{\Delta}$ a privileged search direction.

Training-Free Model Enhancement. RandOpt (Gan & Isola, 2026) shows that isotropic random perturbation *centered at the pretrained model* θ_{base}, combined with top-K ensemble voting, can match fine-tuning; WARM (Ramé et al., 2024) applies weight averaging to RLHF models. RandOpt operates in a different regime from ours (it does not use the fine-tuned checkpoint at all) and requires K-fold ensemble inference. Section 6.3 constructs a *controlled directional ablation* of RandOpt (**Isotropic-at-Endpoint**) that isolates the contribution of the training direction under matched budget.

Loss Landscape Geometry. Li et al. (Li et al., 2018) visualize loss surfaces; SAM (Foret et al., 2021) and SWAD (Cha et al., 2021) seek flat minima; Cohen et al. (Cohen et al., 2021) characterize the edge of stability. These works focus on *during-training* optimization. Our Hessian analysis complements them by characterizing the *post-training* landscape, revealing a geometrically privileged "sharp valley" along the training direction.

3 Problem Formulation

Given compatible start and end checkpoints θ_{start} and θ_{end}, we aim to find $\theta^* = \theta_{\text{end}} + \delta$ that improves downstream evaluation, *without computing gradients*. We define the **endpoint displacement**

$$\mathbf{\Delta} := \theta_{\text{end}} - \theta_{\text{start}}. \tag{1}$$

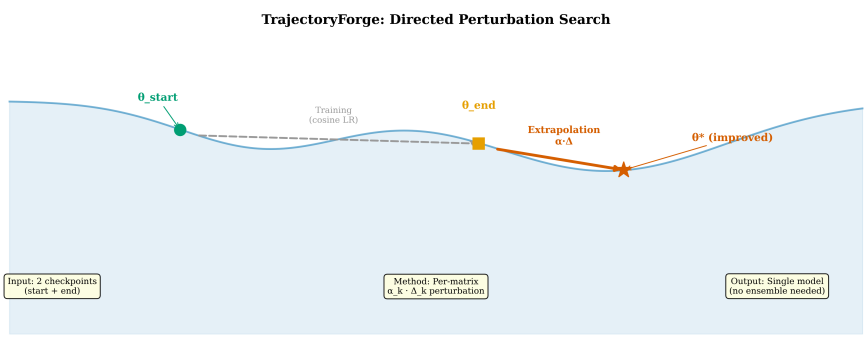


Figure 1: **TrajectoryForge overview.** From only two published checkpoints θ_{start} (pre-fine-tuning) and θ_{end} (fine-tuned) we compute a single training direction $\mathbf{\Delta} = \theta_{\text{end}} - \theta_{\text{start}}$ and decompose it into $K = 291$ per-matrix components $\mathbf{\Delta}_k$. For each candidate j we draw per-matrix scaling factors $\alpha_k \sim |\mathcal{N}(0, \sigma)|$ from a seeded RNG, form $\theta_j = \theta_{\text{end}} + \sum_k \alpha_k \mathbf{\Delta}_k$, and score it on a small evaluation set. Only the triples $(seed, \sigma, s_j)$ are stored; the winning candidate is reconstructed deterministically from its seed. The entire pipeline uses zero gradient computation and deploys as a *single* model with no inference-time overhead.

For plain SGD, this displacement can be expressed as an accumulated sequence of parameter updates. With momentum, Adam-style preconditioning, weight decay, clipping, or multi-stage post-training, however, it is not the raw sum of gradients; our method requires neither that identity nor access to optimizer state.

The displacement is a deterministic, reproducible function of two tensors and can retain structure shared across their checkpoint lineage. For our internally trained pair, the lineage is directly observed. For publicly released base→instruct pairs, the exact optimizer, schedule, intermediate trajectory, and sometimes the full post-training lineage are unavailable; we therefore interpret $\mathbf{\Delta}$ conservatively as a checkpoint displacement rather than a fully observed training trajectory. This distinction bounds our mechanistic claims while leaving the algorithm unchanged.

Checkpoint scope. The seven primary pairs have two provenance regimes. For DeepSeek-Coder-6.7B, the start and end checkpoints come from our observed SFT run. The other six pairs are publicly released base→instruct checkpoints for which the optimizer, schedule, intermediate states, and full post-training lineage are not available. We apply the same subtraction and extrapolation rule in both regimes, but reserve optimizer- and schedule-based interpretations for settings where those facts are known. For the released pairs, the empirical object under study is precisely the endpoint displacement between compatible published checkpoints; we do not assume that an arbitrary long-range displacement, such as one from random initialization, remains useful for local extrapolation.

4 Method

4.1 Directed Perturbation Search

The global alpha scan (Section 5) treats all weight matrices identically, but fine-tuning affects different layers heterogeneously: early layers learn general representations while later layers specialize for the task (Hu et al., 2022). To capture this heterogeneity, we decompose $\mathbf{\Delta}$ into per-matrix components. For a model with K weight matrices $\{W_k\}_{k=1}^{K}$, define $\mathbf{\Delta}_k = W_k^{\text{end}} - W_k^{\text{start}}$. Our method constructs candidate models:

$$\theta'(\boldsymbol{\alpha}) = \theta_{\text{end}} + \sum_{k=1}^{K} \alpha_k \cdot \mathbf{\Delta}_k^{(\text{pad})} \tag{2}$$

where $\alpha_k = |\mathcal{N}(0, \sigma)|$ are per-matrix non-negative scaling factors drawn independently for each weight matrix, and $\mathbf{\Delta}_k^{(\text{pad})}$ embeds $\mathbf{\Delta}_k$ into the full parameter space. This formulation extends Task Arithmetic (Ilharco et al., 2023) from a single global α to K-dimensional per-matrix scaling ($K =$

291 for 7B models), unlocking substantially richer exploration of the weight space while keeping the search space tractable (from $d \sim 10^9$ to $K \sim 10^2$). The complete algorithm is given in Algorithm 1.

Algorithm 1 Directed Perturbation Search (TrajectoryForge)

Require: θ_{start}, θ_{end}, σ values, N samples
Compute $\mathbf{\Delta}_k = W_k^{\text{end}} - W_k^{\text{start}}$ for all K matrices
for $j = 1$ to N **do**
  Set RNG seed $= j$
  for $k = 1$ to K **do**
    $\alpha_k \leftarrow |\mathcal{N}(0, \sigma)|$ {Non-negative, continue training direction}
  end for
  $\theta_j \leftarrow \theta_{\text{end}} + \sum_k \alpha_k \cdot \mathbf{\Delta}_k^{(\text{pad})}$
  $s_j \leftarrow \text{Evaluate}(\theta_j)$
  Store (j, σ, s_j); discard θ_j
end for
return $\arg\max_j s_j$ {Reconstruct via seed}

Key design choices. (1) *Per-matrix granularity* ($K = 291$ for 7B models) reduces the search space from $d \sim 10^9$ to $K \sim 10^2$ while preserving intra-matrix coherence—motivated by the empirical observation that attention projections and later layers change disproportionately during fine-tuning. (2) *Non-negative* α restricts extrapolation to the forward direction; negative α undoes fine-tuning progress (Fig. 2). (3) *Half-normal distribution* concentrates scaling near zero (median $\approx 0.67\sigma$) while allowing occasional larger values, giving exploration-exploitation trade-off via a single σ. (4) *Seed-based storage*: only $(seed, \sigma, score)$ triples are kept (a few KB total), enabling deterministic reconstruction without storing model weights.

Reconstruction semantics. A candidate is identified by its checkpoint pair, perturbation scale σ, and RNG seed. Reconstruction recomputes each matrix displacement $\mathbf{\Delta}_k = W_k^{\text{end}} - W_k^{\text{start}}$, regenerates the seeded coefficients $\alpha_k = |\mathcal{N}(0, \sigma)|$, and applies Eq. (2) to θ_{end}. Thus the search need not retain every candidate checkpoint: the stored $(seed, \sigma, score)$ record identifies which sampled coefficients to regenerate, while the two boundary checkpoints supply the weights and displacement. The score is retained for ranking rather than used in reconstruction, and the selected candidate deploys as one model rather than an ensemble.

Computational cost. TrajectoryForge avoids gradient computation entirely: a one-time $O(d)$ preprocessing pass computes all $\mathbf{\Delta}_k$, and each of the N candidates requires only an inference-only forward evaluation, replacing four to five orders of magnitude of gradient computation (detailed analysis in Appendix C).

5 Mechanism Analysis

We provide an *empirical* characterization of *why* extrapolation along $\mathbf{\Delta}$ improves fine-tuned models, rather than a formal proof. The analysis proceeds in two steps: (i) an inverted-U alpha scan that identifies a locally suboptimal endpoint, and (ii) Hessian curvature diagnostics showing that $\mathbf{\Delta}$ is a geometrically atypical direction at θ_{end}. We phrase all claims as empirically supported hypotheses; the underlying quantities ($\mathbf{\Delta}$ direction, curvature ratio, spectrum) are all estimated from finite checkpoints and finite calibration batches, and rigorous derivations from optimizer dynamics remain open.

Why extrapolation can work. For a known run with cosine decay $\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\pi t/T))$, late-stage step sizes approach $\eta_{\min}$, motivating the hypothesis that the endpoint may remain improvable along its accumulated update direction. For released pairs with unknown schedules, this is a motivation rather than an established cause. Empirically, the Qwen-Coder-1.5B alpha scan (Fig. 2) yields $\alpha^* = 0.05$ with HE $= 0.720$ (+3.1 percentage points); per-matrix search reaches 0.738 (7.1% relative improvement). Both signs of small α improve over the baseline, supporting local endpoint suboptimality more directly than a uniquely forward descent direction; large α eventually degrades performance.

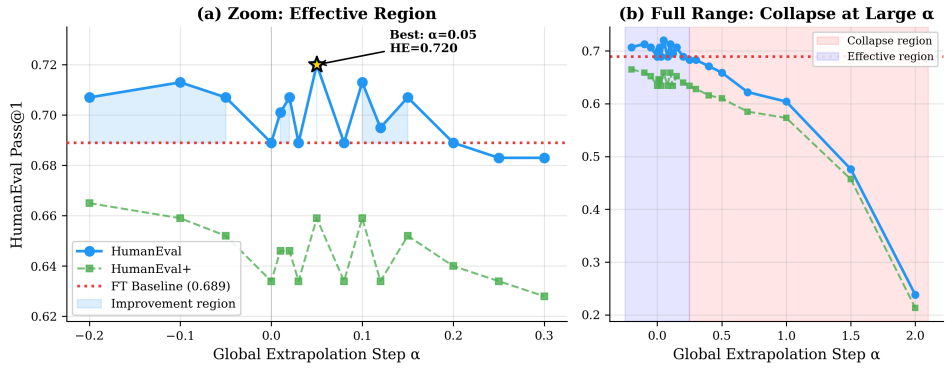


Figure 2: **Alpha scan on Qwen-Coder-1.5B** ($\theta' = \theta_{\text{end}} + \alpha \cdot \mathbf{\Delta}$). **Left**: inverted-U with peak at $\alpha^* = 0.05$ (HE=0.720, +3.1 pp). **Right**: graceful degradation beyond $\alpha = 1.5$. Per-matrix search reaches HE=0.738 (7.1% relative).

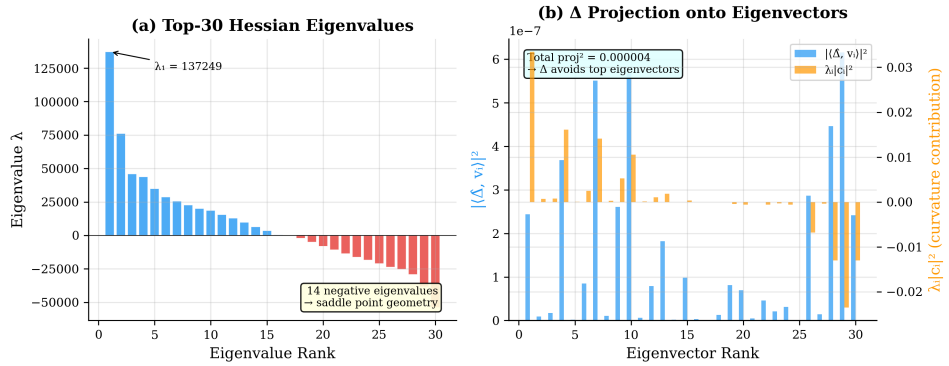


Figure 3: **Hessian spectrum and $\hat{\mathbf{\Delta}}$ projection on Qwen-Coder-1.5B.** Top-30 eigenvalues (both signs) reveal a saddle-point geometry at θ_{end} rather than a local minimum; the fine-tuning direction $\hat{\mathbf{\Delta}}$ projects near zero onto all top eigenvectors ($< 10^{-6}$), placing it in an atypical low-projection subspace that avoids the sharpest curvature while still producing meaningful loss change (372× larger curvature than random directions). This "low-projection, high-directional-curvature" pattern is the geometric signature that isotropic Gaussian perturbation cannot reproduce.

Why the endpoint displacement may be informative. Let $H = \nabla^2 \mathcal{L}(\theta_{\text{end}})$. Our finite-batch diagnostics show atypical geometry along $\mathbf{\Delta}$: (i) $\hat{\mathbf{\Delta}}^\top H \hat{\mathbf{\Delta}}$ exceeds the random-direction estimate by 372× on Qwen-Coder-1.5B and by $> 10\times$ on two additional models; (ii) $\mathbf{\Delta}$ has low projection onto the estimated top eigenvectors ($< 10^{-6}$); and (iii) the estimated spectrum includes large negative eigenvalues. These observations offer a possible geometric explanation for the higher success rate of directed search in our matched control, but do not establish causality. Full estimation details and per-model results are in Appendix E.

6 Experiments

6.1 Implementation Details

Models and datasets. We evaluate on 7 model pairs spanning three families and three scales (1.5B, 7B, 14B), covering code generation (DeepSeek-Coder-6.7B (Guo et al., 2024), Qwen2.5-Coder-{1.5B,7B,14B} (Hui et al., 2024), CodeLlama-7B-Python (Rozière et al., 2023)), mathematical reasoning (Qwen2.5-Math-{1.5B,7B} (Yang et al., 2024)), and instruction/knowledge (Qwen2.5-7B Base→Instruct (Qwen Team, 2024)). We focus on these domains because they offer *objective, deterministic evaluation*—code via execution-based test suites and math via ground-truth

Table 1: **TrajectoryForge results on code/math tasks (7 pairs).** †: mean $>$ FT with $p < 10^{-20}$. Best σ^* per model: 0.005–0.15 (see Appendices F and H).

Model	FT	Mean	Best	%>FT
Code Generation (HumanEval)				
DeepSeek-Coder-6.7B	.732	**.744**†	**.774**	81.2
Qwen-Coder-1.5B	.689	**.706**†	**.738**	90.0
Qwen-Coder-7B	.909	.900	**.921**	11.9
Qwen-Coder-14B	.921	.911	.921	0.0
CodeLlama-7B-Py	.415	.276	**.433**	1.3
Math Reasoning (GSM8K)				
Qwen-Math-1.5B	.842	.838	**.855**	28.8
Qwen-Math-7B	.945	.943	**.948**	10.0

answers—providing reliable signal for detecting small improvements. For DeepSeek-Coder-6.7B we use our own SFT checkpoint; the other 6 pairs use publicly released base→instruct pairs. Benchmarks: HumanEval (Chen et al., 2021) (164 problems, pass@1) via EvalPlus (Liu et al., 2023), GSM8K (Cobbe et al., 2021), MBPP (Austin et al., 2021) (held-out), IFEval (Zhou et al., 2023), and MMLU (Hendrycks et al., 2021) (5-shot). Full FT baselines and per-model details are in Table 3 (Appendix B).

Sampling budget. We use $N = 200$ candidates per (σ, model) configuration. Per-model totals in Table 1 ($N = 170$–480) reflect (i) $\sim$10–20% vLLM OOM/timeout attrition (deterministic under identical seeds) and (ii) extended σ sweeps for smaller models where inference is cheap. All Isotropic-at-Endpoint comparisons use *identical* σ grid and target N, so success rates and mean gaps are computed on matched samples. Per-σ breakdowns in Appendix H.

6.2 Main Results

Table 1 shows that the *best sampled candidate* matches or exceeds the endpoint on 6 of 7 code/math pairs. The distributional evidence is narrower: only Qwen-Coder-1.5B (90.0% $>$ FT) and DeepSeek-Coder-6.7B (81.2% $>$ FT; $N = 170$ valid candidates) improve at the **mean** level with $p < 10^{-20}$. The remaining gains depend on best-of-N selection, including CodeLlama, whose best improves while its mean declines substantially. Results across independently released checkpoint pairs provide cross-model breadth, not multi-seed replication. Reported headline gains are relative unless explicitly marked as percentage points.

Distributional evidence. For Qwen-Coder-1.5B, 90.0% of candidates exceed the endpoint, the mean rises from 0.689 to 0.706, and the 95% confidence interval for the mean difference is $[+0.015, +0.018]$ ($N = 480$). For DeepSeek-Coder-6.7B, 81.2% of the $N = 170$ valid candidates exceed the endpoint, the mean rises from 0.732 to 0.744, and the corresponding interval is $[+0.008, +0.016]$. Both tests give $p < 10^{-20}$; complete statistics, including the five non-improving mean cases, appear in Appendix I. These estimates characterize the sampled candidate distributions and should not be interpreted as multi-seed training replication.

Comparison with existing methods. On Qwen-Coder-1.5B: Task Arithmetic (Ilharco et al., 2023) and ExPO (Zheng et al., 2024) (global α scan) achieve +3.1%; Isotropic-at-Endpoint (our controlled variant of RandOpt (Gan & Isola, 2026), matched budget) matches the best score on DeepSeek-6.7B but with only 41% success rate (vs. our 81%) and $50\times$ inference cost when used with the ensemble voting of the original RandOpt; Model Soups (Wortsman et al., 2022) requires multiple training runs unavailable in our setting. Our method achieves +7.1% with $2\times$ the success rate and no ensemble.

Direction-task alignment. We apply directed perturbation to Qwen2.5-7B (Base→Instruct) on two contrasting benchmarks. The best candidate reaches **80.0%** on IFEval (+1.1 percentage points), while MMLU changes remain within ± 0.15 percentage points across the tested settings. We interpret

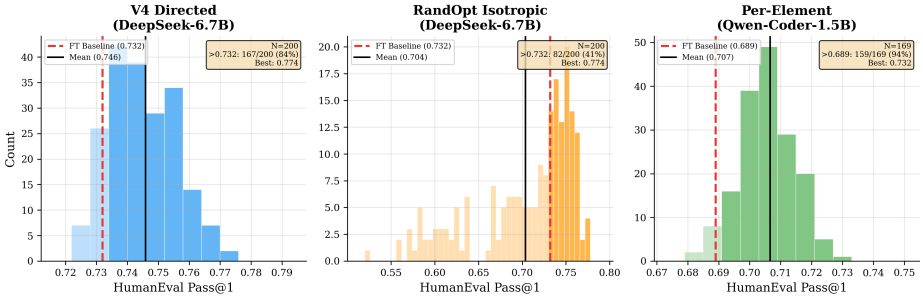


Figure 4: **Score distributions under matched settings. Left**: TrajectoryForge on the $N = 170$ valid DeepSeek-6.7B candidates (81.2% $>$ FT). **Center**: Isotropic-at-Endpoint (41% $>$ FT in its matched comparison). **Right**: per-element TrajectoryForge on Qwen-Coder-1.5B at $\sigma = 0.02$ (93% $>$ FT; $N = 40$). The controlled comparison changes the perturbation direction while retaining the endpoint center and evaluation protocol.

this as no practically meaningful MMLU change, rather than a formal non-inferiority result. The contrast suggests—but does not prove—that a small Base→Instruct benchmark gap predicts limited benefit (Appendix G).

6.3 Directed vs. Random Perturbation: Isolating the Directional Contribution

A natural concern is whether TrajectoryForge's gains stem from the endpoint displacement or merely from best-of-N search near θ_{end}. We therefore construct **Isotropic-at-Endpoint**, $\theta' = \theta_{\text{end}} + \sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, using the same endpoint center, target sample budget, σ schedule, and evaluation protocol; valid-sample totals reflect the attrition described in Section 6.1.

Figure 4 shows a right-shifted directed distribution: 81.2% of the $N = 170$ valid DeepSeek candidates exceed FT, versus 41% for the isotropic control. On Qwen-Coder-1.5B, the per-element setting at $\sigma = 0.02$ succeeds on 93% of $N = 40$ candidates. These controls provide evidence that displacement-aligned perturbations are more reliable than neighborhood search alone under the tested settings; they do not establish that direction is the sole causal factor.

The angular sweet spot. Sweeping the equivalent angular deviation θ between the applied perturbation and $\mathbf{\Delta}$ (full sweep in Appendix K) reveals that the success rate peaks sharply at $\theta \approx 22°$ (**93%**)—a small stochastic dispersion around $\mathbf{\Delta}$—versus 43% for the pure direction (0°, the global alpha scan) and 41% for fully random perturbation ($\sim 90°$). This gives a principled reason to prefer per-matrix half-normal α_k over both the ExPO-style global scalar and full isotropic perturbation.

Hessian validation across models. The sharp-valley prediction generalizes: RLHF-trained models show sharp-valley directions ($r \gg 1$, up to 372×) favoring small σ^*, while pure-SFT models show flat-plateau directions ($r < 1$) favoring larger σ^*; σ^* also decreases with model size (Appendices E, F, and H).

6.4 Cross-Benchmark Selection and Evaluation

We evaluate all 529 Qwen-Coder-1.5B candidates on HumanEval and MBPP (Austin et al., 2021) (378 problems). This permits two directional cross-benchmark tests: the candidate selected solely by MBPP (seed=374, $\sigma = 0.15$) reaches HumanEval 0.713 (+2.4 percentage points) without using HumanEval for selection; conversely, an HE-only selection can be evaluated on MBPP. We also report a joint-selected candidate (seed=153), but because it uses both benchmarks it is not a held-out result. Across all candidates, 90.9% exceed the HumanEval baseline. Together with mean-level gains for 2 of 7 models, these results provide evidence against purely benchmark-specific selection effects, while not eliminating best-of-N bias (Appendix F).

Table 2: **Cross-benchmark results** (Qwen-Coder-1.5B, 529 candidates; per-σ results in Appendix F). MBPP-only selection transfers to HE (+2.4 pp). Joint selection reaches HE 0.738 and MBPP 0.696 but is not a held-out test.

Selection	σ	HE	MBPP	ΔHE (pp)	ΔMBPP (pp)
FT baseline	—	.689	.696	—	—
MBPP-selected	0.15	**.713**	.704	+2.4	+0.8
Joint-selected	0.15	**.738**	.696	+4.9	±0.0
Distribution (mean)	mix	.707	.684	+1.8	−1.2

Selection protocols. These results separate three uses of benchmark scores. Same-benchmark best-of-N shows that the sampled set contains an improved candidate but retains selection bias. The MBPP-selected candidate does not use HumanEval scores, so HumanEval is held out for that rule; the joint-selected candidate uses both benchmarks and is held out on neither. For deployment, a fixed candidate should therefore be checked on an independent validation benchmark. Our cross-benchmark result demonstrates one instance of transfer, not a guarantee that every selected gain will generalize.

What the statistics estimate. The mean and confidence interval describe the tested candidate distribution; success rate estimates how often one draw exceeds the endpoint; and the maximum reports the result after the stated search budget. Summaries use valid evaluations. Because failures may depend on perturbation scale, these aggregates are conditional on successful evaluation. We therefore report valid sample counts, distinguish selected gains from average improvements, and treat cross-benchmark evaluation as an additional check rather than proof that selection bias is absent.

6.5 When TrajectoryForge Does Not Help: A Predictable Regime

Table 1 also identifies important failure modes. For Qwen-Coder-14B, no candidate exceeds FT: the best ties 0.921, while the candidate mean declines to 0.911. For MMLU, changes stay within ± 0.15 percentage points across tested settings. Thus a best-of-N search is not intrinsically safe and requires a separate validation criterion.

Direction alone is insufficient. The start-small ablation applies the same displacement from θ_{start} using $\alpha \in [0, 0.2]$ on DeepSeek-Coder-6.7B. None of the 200 samples exceeds the 0.732 endpoint: the best reaches 0.677 and the mean is 0.639, compared with 0.774 best and 0.744 mean when perturbing from θ_{end}. Thus the method is a local post-endpoint search using a structured displacement, not evidence that $\mathbf{\Delta}$ is globally improving throughout weight space (Appendix D.1).

Tentative pre-hoc heuristic. In our experiments, small Base→Instruct gaps coincide with limited benefit: the same Qwen2.5-7B displacement improves IFEval, where the endpoint gap is larger, but changes MMLU little. This suggests scoring both checkpoints before search and skipping settings with little measurable headroom. The thresholds observed here are empirical and should not be treated as universal.

6.6 Alternative Explanations Considered

Three alternatives available to the original trainer—fixing the learning-rate schedule, resuming fine-tuning, or averaging the last K checkpoints (SWA)—require artifacts that released models do not ship; Appendix L discusses each and explains why random search replaces gradient-based σ optimization, and Appendix M gives a three-step practitioner protocol. The fourth alternative—best-of-N selection bias—is addressed directly here.

Is the improvement just best-of-N selection bias? Three analyses mitigate, but do not eliminate, this concern: (i) Qwen-Coder-1.5B and DeepSeek-6.7B improve at the *mean* level ($p < 10^{-20}$); (ii) MBPP-only selection transfers to HumanEval by +2.4 percentage points; and (iii) Isotropic-at-Endpoint controls for neighborhood search under matched settings. The other model pairs still rely on best-of-N selection and therefore require independent validation.

7 DISCUSSION

Our results paint a coherent picture of post-training optimization, organized around three converging lines of evidence.

Endpoint suboptimality is observable but not universal. The alpha scan shows an inverted-U curve peaking beyond one endpoint, while best-of-N gains occur on six model pairs. Because the public pairs lack complete optimizer and schedule records, these results establish exploitable checkpoint displacement rather than cosine decay as a universal cause.

Geometry offers a hypothesis, not a proof. Finite-batch Hessian estimates show high directional curvature and low top-eigenspace projection for some models. This pattern is consistent with the matched-control advantage of directed search, but varies across models and does not establish causality.

Selection remains central. Only two pairs improve at the mean level; the others depend on best-of-N selection, and the 14B candidate mean declines. Cross-benchmark results and the IFEval–MMLU contrast suggest when displacement-aligned search may transfer, but deployment should use a validation benchmark not used to select the final candidate.

Operational implication. TrajectoryForge should be viewed as a search-and-validation procedure rather than a deterministic transformation that always improves a checkpoint. A pilot can localize σ before concentrating the inference budget. The winner should then be frozen and evaluated on an independent benchmark; if that check degrades, the original endpoint remains the appropriate deployment choice.

Search-budget interpretation. Increasing N raises both the chance of finding a tail candidate and the risk of adapting to finite benchmark noise. Comparisons should therefore control the candidate budget and report the mean and success rate alongside the maximum. Performance as a function of N would further expose the marginal return from additional search.

Evaluation-set reuse. Ranking many candidates on the same finite benchmark can specialize the selected model to that evaluation set even without gradients. Cross-benchmark transfer and positive candidate means mitigate this concern, but do not replace fresh confirmation data. A stronger protocol would select σ and seed on a search split, freeze the candidate, and evaluate it once on a disjoint confirmation split. Reporting this split together with the number of attempted and valid candidates would make the effective optimization budget explicit and prevent later evaluation choices from leaking back into candidate selection.

Open questions. A key challenge is *predicting σ^* without exhaustive search*. The empirical trend ($\sigma^* \propto 1/\text{model size}$) provides initial guidance, but a closed-form relationship between Hessian spectrum and optimal perturbation magnitude remains elusive; lightweight Hessian probes could potentially serve as a cheap proxy. A second question is whether coefficients can be estimated from checkpoint statistics before evaluating any candidates. Layer-wise displacement norms, curvature proxies, or activation changes may provide inexpensive priors over useful perturbation scales. Such predictors would reduce the search budget while preserving the gradient-free setting, and would turn the current empirical scale-selection procedure into a more transferable rule across model families and tasks. Evaluating these predictors prospectively on unseen model families, model scales, downstream tasks, and independently collected evaluation sets is an important next step for establishing their reliability, generality, and practical value.

8 CONCLUSION

We presented TrajectoryForge, a gradient-free framework that extrapolates the displacement between two compatible checkpoints. Across seven model pairs, the best candidate matches or exceeds the endpoint for six, while two improve at the mean level. Hessian and cross-benchmark analyses characterize the useful regime, but best-of-N selection still requires independent validation. Limitations are detailed in Appendix A.

AI USE STATEMENT

We used generative AI tools for software-code assistance and for drafting, polishing, and formatting portions of the manuscript. We did not use them to propose the method, formulate research hypotheses or mathematical claims, design or execute experiments, generate synthetic datasets, or interpret experimental results. The authors reviewed and verified all AI-assisted code and text and take responsibility for the final manuscript, claims, results, and accompanying artifacts.

ETHICS STATEMENT

This work uses only publicly available model checkpoints and public benchmarks (HumanEval, MBPP, GSM8K, IFEval, MMLU); it involves no human subjects, no personally identifiable information, and no new data collection. The method is not designed to introduce qualitatively new capabilities; however, weight-space perturbations may alter model behavior, so selected models should be evaluated for capability and safety changes before deployment. We acknowledge the ICLR Code of Ethics and identify no conflicts of interest specific to this submission.

REPRODUCIBILITY STATEMENT

We make the following efforts to ensure reproducibility. The method is fully specified by Algorithm 1 and Eq. (2); all hyperparameters (σ grids, sample budgets, seeds) are reported in Section 6.1 and Appendix H, and candidate models are deterministically reconstructible from their stored seeds. Evaluation benchmarks and protocols are described in Section 6.1, with per-model details in Appendix B and complete per-σ results in Appendices F–H. Anonymized source code is included in the supplementary material.

REFERENCES

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: Domain generalization by seeking flat minima. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

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

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. DeepSeek-Coder: When the large language model meets programming – the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

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*, 2022.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

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

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

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Hwang. Model stock: All we need is just a few fine-tuned models. In *European Conference on Computer Vision*, 2024.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhenyi Lu, Chenghao Jiang, Jianwei He, Weiwen Xiao, Bin Zhu, Lingyu Liu, Jie Jiang, and Min Chen. Twin-merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

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

Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Olivier Bachem, and Johan Ferret. WARM: On the benefits of weight averaged reward models. In *International Conference on Machine Learning*, 2024.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code Llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak 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*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 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*, volume 36, 2023.

An Yang, Beichen Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Weak-to-strong extrapolation expedites alignment. In *International Conference on Machine Learning*, 2024. arXiv:2404.16792.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Limitations and Scope of Validity

Limitations. (1) Evaluating N candidates incurs inference cost $\propto N$ and requires a selection set. (2) Only two of seven pairs improve at the mean level; the others depend on best-of-N selection, and the 14B candidate mean is lower than its endpoint. Cross-benchmark evaluation mitigates but does not eliminate selection bias. (3) Optimal σ varies across models, so scale trends and the proposed pre-hoc heuristic require broader validation. (4) For public checkpoint pairs, the exact lineage, optimizer, and schedule are not fully observed; $\mathbf{\Delta}$ should therefore be interpreted as an endpoint displacement rather than a recovered raw-gradient trajectory.

Scope of validity. Our experiments concern compatible pretrained and post-trained checkpoints whose displacement is small relative to the pretraining trajectory. We do *not* claim that $\mathbf{\Delta}$ is useful for arbitrary optimization: a displacement from random initialization to a final solution need not point toward further improvement, and heavy feature-learning regimes may invalidate local linear extrapolation. For public pairs, compatibility is inferred from released checkpoint families rather than a fully observed single-run trajectory; consequently, optimizer- and schedule-specific explanations remain hypotheses.

B Model Pairs and Baselines

Table 3: Model pairs used in experiments. The last two rows use the same model pair evaluated on different benchmarks to test direction-task alignment.

Model Pair	Task	Params	Benchmark	FT Baseline
DeepSeek-Coder-6.7B (SFT) (Guo et al., 2024)	Code	6.7B	HumanEval	0.732
Qwen2.5-Coder-1.5B → Instruct (Hui et al., 2024)	Code	1.5B	HumanEval	0.689
Qwen2.5-Coder-7B → Instruct (Hui et al., 2024)	Code	7B	HumanEval	0.909
Qwen2.5-Coder-14B → Instruct (Hui et al., 2024)	Code	14B	HumanEval	0.921
CodeLlama-7B → Python (Rozière et al., 2023)	Code	7B	HumanEval	0.415
Qwen2.5-Math-1.5B → Instruct (Yang et al., 2024)	Math	1.5B	GSM8K	0.842
Qwen2.5-Math-7B → Instruct (Yang et al., 2024)	Math	7B	GSM8K	0.945
Qwen2.5-7B → Instruct (Qwen Team, 2024)	Instruct	7B	IFEval	0.789
Qwen2.5-7B → Instruct (Qwen Team, 2024)	Knowledge	7B	MMLU	0.742

C Computational Cost Analysis

Computational cost. TrajectoryForge avoids gradient computation entirely, and its cost decomposes into two parts. (i) A *one-time* preprocessing step computes $\mathbf{\Delta}_k = W_k^{\text{end}} - W_k^{\text{start}}$ for each of the

$K = 291$ matrices—a single pass over the two checkpoints with cost $O(d)$ scalar subtractions and no autograd graph. (ii) Each of the N candidates requires a forward-pass evaluation on the target benchmark; no candidate is ever materialized on disk—only the tuple $(seed, \sigma, s_j)$ is stored (a few KB total), and the winning model is reconstructed deterministically at the end. Compared with resuming fine-tuning for even a single additional epoch, which would require $O(T_{\text{data}} \cdot d)$ backward passes plus optimizer state, TrajectoryForge replaces roughly four to five orders of magnitude of gradient computation with a linear number of *inference-only* evaluations, and unlike RandOpt's top-K ensemble voting (Gan & Isola, 2026), the deployed model is a *single* network with *no* inference-time overhead.

D Ablation Studies

D.1 Start-Small: Direction Without Position

To test whether the training *direction* alone suffices (without reaching θ_{end}), we evaluate $\theta' = \theta_{\text{start}} + \alpha \cdot \mathbf{\Delta}$ with small $\alpha \in [0, 0.2]$.

Table 4: Start-small ablation on DeepSeek-Coder-6.7B. Direction alone is insufficient; position matters too.

Experiment	Best HE	Mean HE	% > FT (0.732)
Directed (from θ_{end})	**0.774**	0.744	81.2%
Start-small (from θ_{start})	0.677	0.639	0.0%

Finding: 0/200 start-small samples exceed the FT baseline. The training direction is necessary but not sufficient—the model must first reach the vicinity of θ_{end} before extrapolation becomes effective.

D.2 Per-Element vs. Per-Matrix Perturbation

Table 5: Per-element vs. per-matrix perturbation on Qwen-Coder-1.5B (40 samples per σ).

Method	Granularity	Best HE	Mean HE	% > FT
Per-matrix (σ=0.02)	$K = 291$	0.726	0.703	95.0%
Per-element (σ=0.02)	$K = d$	**0.732**	**0.718**	**93.0%**
Per-element (σ=0.05)	$K = d$	0.720	0.700	75.0%
Per-element (σ=0.10)	$K = d$	0.713	0.685	92.0%

Per-element perturbation at $\sigma = 0.02$ achieves 93% success rate with the highest mean score (0.718), while per-matrix achieves 95% success rate but lower mean (0.703). Both significantly outperform the FT baseline.

E Hessian Analysis Details

E.1 Computational Setup

For each model pair, we compute the Hessian-vector product (HVP) using numerical differentiation ($\varepsilon = 0.001$) with 2 calibration batches. We use Lanczos iteration (30 steps) for eigenvalue decomposition (1.5B models only; 7B+ models exceed the `torch.dot` 32-bit index limit or GPU memory), random direction sampling (30 directions) for curvature ratio estimation, and Hutchinson trace estimator (30 samples) for $\text{tr}(H)$ estimation. The eigenvalue spectrum and $\hat{\mathbf{\Delta}}$-projection visualization for Qwen-Coder-1.5B are shown in the eigenvalue-spectrum figure in the main paper.

E.2 Complete Hessian Results

Table 6: Complete Hessian spectral analysis results. z: z-score of training direction curvature against the random baseline distribution. Lanczos eigenvalue decomposition was infeasible for 7B+ models due to `torch.dot` 32-bit index overflow ($d > 2^{31}$) and GPU memory constraints.

Model	$\hat{\Delta}^\top H \hat{\Delta}$	$\mathbb{E}[\hat{\epsilon}^\top H \hat{\epsilon}]$	**Ratio**	z	$\lambda_{\max}$	$\lambda_{\min}$	$\|\text{proj}\|^2$
Qwen-Coder-1.5B	0.0191	0.00005	372×	2.78	137,249	−54,011	6.4×10^{-7}
Qwen-Coder-7B	0.0130	0.00103	12.7×	2.78	*infeasible ($d > 2^{31}$)*		
Qwen-Math-1.5B	−0.0008	−0.00007	11.9×	1.73	49,617	−50,702	5.9×10^{-8}
Qwen-Math-7B	0.0008	−0.00055	−1.46×	0.69	*infeasible ($d > 2^{31}$)*		
DeepSeek-Coder-6.7B	0.0006	0.00326	0.17×	−0.26	*infeasible ($d > 2^{31}$)*		
CodeLlama-7B-Python	0.0081	0.03726	0.22×	−0.11	*infeasible ($d > 2^{31}$)*		

F Hyperparameter Sensitivity and Cross-Benchmark Evaluation

The primary hyperparameter in TrajectoryForge is the perturbation scale σ. We analyze its sensitivity on Qwen-Coder-1.5B by evaluating 529 candidates across 9 σ values on both HumanEval and MBPP (Austin et al., 2021) (378 problems, plus MBPP+). A benchmark is held out only for selection rules that do not use its scores.

Hyperparameter sensitivity. Table 7 reveals that the method is **robust to σ choice**: HumanEval Best ≥ 0.720 for all $\sigma \in [0.005, 0.30]$, and HumanEval Mean remains above the FT baseline (0.689) for all tested values. The optimal range is $\sigma \in [0.10, 0.20]$, with performance degrading gracefully outside this range. On the held-out MBPP benchmark, MBPP Mean stays within 2.9% of the baseline across all σ, confirming that the method does not catastrophically degrade on unseen tasks regardless of σ choice.

Cross-benchmark selection. The candidate selected solely by MBPP (seed=374, $\sigma = 0.15$) reaches HumanEval 0.713 (+2.4 percentage points) without using HumanEval scores. The candidate selected jointly by $(\text{HE} + \text{MBPP})/2$ (seed=153) reaches HE 0.738 and MBPP 0.696, but is not held out with respect to either benchmark. The mean HumanEval score over all 529 candidates is 0.707 (+1.8 percentage points), and 90.9% exceed the HumanEval baseline. These are complementary distributional and cross-selection results, not a single strict held-out protocol.

Table 7: **Cross-benchmark evaluation on MBPP/MBPP+** (Qwen-Coder-1.5B, 529 candidates). The MBPP-selected model transfers to HE=0.713 (+2.4 pp without HE selection). The joint-selected model reaches HE=0.738 and MBPP=0.696, but is not a held-out result.

σ	**N**	**MBPP Mean**	**MBPP+ Mean**	**HE Mean**	**HE Best**	**HE>FT**	**MBPP≥FT**
0.005	40	0.685	0.591	0.699	0.713	90.0%	2.5%
0.01	40	0.687	0.592	0.702	0.720	95.0%	5.0%
0.02	87	0.690	0.595	0.704	0.726	95.4%	11.5%
0.05	87	0.689	0.594	0.702	0.726	74.7%	6.9%
0.10	87	0.685	0.591	0.709	0.732	90.8%	9.2%
0.15	87	0.681	0.590	0.711	**0.738**	97.7%	2.3%
0.20	87	0.676	0.585	0.713	0.738	93.1%	0.0%
0.25	7	0.671	—	0.723	0.738	100.0%	0.0%
0.30	7	0.675	—	0.716	0.732	100.0%	0.0%
All	529	0.684	0.591	0.707	0.738	90.9%	5.5%

G Direction-Task Alignment: IFEval and MMLU Details

Table 8: **IFEval per-σ results** (Qwen2.5-7B). Instruct baseline: 78.93%. Best model (seed=28, $\sigma = 0.02$) achieves 80.04% (+1.1 percentage points).

σ	N	Mean	Best	$>$ Instruct	Best Seed
0.005	10	0.7854	0.7967	3/10	3
0.010	10	0.7852	0.7948	2/10	14
0.020	**10**	**0.7865**	**0.8004**	**3/10**	**28**
0.050	10	0.7824	0.7911	2/10	35
0.100	10	0.7835	0.7967	1/10	49
All	50	0.7846	0.8004	11/50	—

Table 9: **MMLU per-σ results** (Qwen2.5-7B, 5-shot). Instruct baseline: 74.16%. Mean changes remain within ± 0.15 percentage points; we observe no practically meaningful change but do not claim formal equivalence.

σ	N	Mean	Best	$>$ Instruct	Δ vs Instruct
0.005	10	0.7420	0.7427	7/10	+0.04%
0.010	10	0.7419	0.7428	6/10	+0.03%
0.020	10	0.7415	0.7422	5/10	−0.01%
0.050	10	0.7403	0.7416	4/10	−0.13%
0.100	10	0.7401	0.7413	5/10	−0.15%
All	50	0.7412	0.7428	27/50	−0.04%

Interpretation. The contrast is consistent with task-dependent information in the endpoint displacement: IFEval improves while MMLU changes little. Because these experiments do not establish equivalence or causality, we treat the Base→Instruct benchmark gap only as a tentative heuristic for prioritizing search.

H Per-Sigma Detailed Analysis

Table 10 provides a detailed breakdown of directed perturbation results on DeepSeek-Coder-6.7B across five σ values, each with 40 independent samples (10 for $\sigma = 0.20$).

Table 10: Per-σ results on DeepSeek-Coder-6.7B.

Model	σ	Samples	Mean	Best	% > FT	% > Extrap.
DeepSeek-6.7B	0.02	40	0.736	0.750	70%	5%
	0.05	40	0.749	0.768	95%	35%
	0.10	40	0.738	**0.774**	70%	10%
	0.15	40	0.748	0.768	80%	30%
	0.20	10	0.753	0.774	100%	60%

Interpretation. Unlike Qwen-Coder-1.5B (which favors small σ), DeepSeek-Coder-6.7B benefits from moderate-to-large perturbation scales. At $\sigma = 0.05$, 95% of candidates exceed the FT baseline with a mean of 0.749 (+1.7%). The best individual score (0.774, +4.2%) is achieved at $\sigma \in \{0.10, 0.20\}$. Notably, $\sigma = 0.20$ achieves 100% success rate and 60% of candidates even surpass the global extrapolation optimum. This behavior is consistent with the flat-plateau geometry identified in Table 6 ($r = 0.17\times$): the low directional curvature allows larger perturbations without overshooting, and the broad basin accommodates more aggressive exploration.

I Statistical Significance Analysis

Table 11: **Statistical significance analysis.** Qwen-Coder-1.5B and DeepSeek-Coder-6.7B show highly significant improvements at the *mean* level ($p < 10^{-20}$). Other models show improvement only at the best-of-N level. Per-model N variation (170–480) reflects (i) $\sim$10–20% vLLM OOM/timeout attrition on a fixed target of $N = 200$ per (σ, model), and (ii) extended σ sweeps on 1.5B models where inference is cheap; all Isotropic-at-Endpoint comparisons reported in the main paper use *identical* target N so success rates and mean gaps are matched.

Model	N	Mean	Binom. p	t-test p	Cohen's d	95% CI
Qwen-Coder-1.5B	480	0.706	1.4×10^{-48}***	6.7×10^{-55}***	1.17 (large)	[+0.015, +0.018]
DeepSeek-Coder-6.7B	170	0.744	$< 10^{-20}$***	$< 10^{-20}$***	0.92 (large)	[+0.008, +0.016]
Qwen-Coder-7B	480	0.900	1.00	1.00	-0.66 (medium)	$[-0.009, -0.006]$
Qwen-Coder-14B	280	0.911	1.00	1.00	-2.02 (large)	$[-0.012, -0.011]$
CodeLlama-7B	462	0.276	1.00	1.00	-0.90 (large)	$[-0.137, -0.105]$
Qwen-Math-1.5B	240	0.838	1.00	1.00	-0.75 (medium)	$[-0.025, -0.018]$
Qwen-Math-7B	240	0.943	1.00	1.00	-1.06 (large)	$[-0.007, -0.006]$

J Per-Layer Analysis

We decompose the training direction $\mathbf{\Delta}$ at the granularity of individual layers and module types (attention projections, MLP up/down/gate, layer norms) to understand *where* fine-tuning concentrates its updates and how local geometry varies across the network.

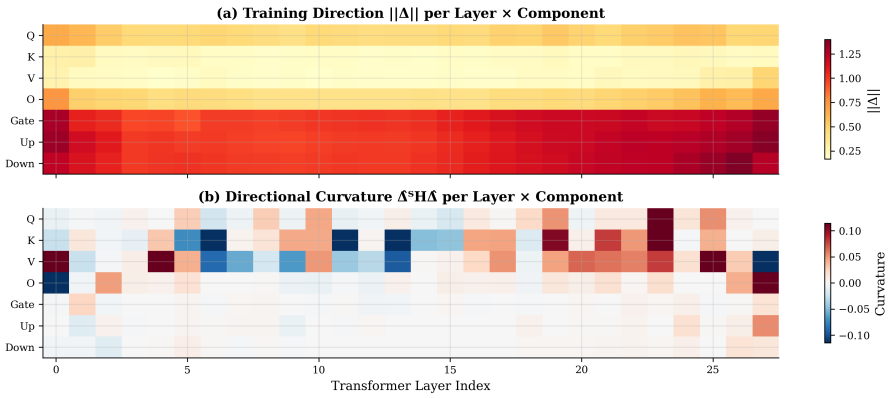


Figure 5: **Training direction anatomy** for Qwen-Coder-1.5B. Top: $\|\boldsymbol{\Delta}\|$; bottom: directional curvature.

Norm–curvature decomposition. Figure 5 jointly displays the $\|\boldsymbol{\Delta}\|$ magnitude (top) and directional curvature $\hat{\Delta}_k^\top H_k \hat{\Delta}_k$ (bottom) for Qwen-Coder-1.5B. While $\|\boldsymbol{\Delta}\|$ is relatively uniform across layers, curvature varies by orders of magnitude, with attention layers exhibiting the sharpest curvature. This decoupling implies that perturbation sensitivity is governed more by local curvature than by update magnitude.

K Angular Deviation Analysis

To quantify how far each perturbation method deviates from the pure training direction, we compute the equivalent angular deviation θ between the applied perturbation vector and $\boldsymbol{\Delta}$. Table 12 presents results on Qwen-Coder-1.5B across the full angular spectrum.

Table 12: Angular deviation spectrum on Qwen-Coder-1.5B.

σ	Equiv. angle θ	Best HE	Success rate	Interpretation
0 (Alpha Scan)	0°	0.720	43%	Pure direction
0.02	21.8°	0.732	93%	Slight deviation
0.05	45.0°	0.720	75%	Moderate deviation
0.10	63.4°	0.713	92%	Large deviation
Isotropic-at-Endpoint	$\sim 90^\circ$	0.774	41%	Fully random

Interpretation. The results reveal a nuanced picture: (1) *Pure direction* (0°, Alpha Scan) yields a moderate best score but only 43% success rate, suggesting the global direction is helpful but not sufficient. (2) *Slight deviation* ($\sim 22^\circ$, $\sigma = 0.02$) achieves the highest success rate (93%), indicating that small per-matrix perturbations around the training direction reliably find better solutions. (3) As deviation increases toward 90° (Isotropic-at-Endpoint), the success rate drops back to 41%, confirming that the training direction provides essential guidance. The sweet spot lies at small angular deviations, where the method benefits from both the directional signal and the stochastic exploration of the per-matrix coefficient space.

L Alternative Explanations: Extended Discussion

Why not fix the learning-rate schedule? Adjusting the schedule requires access to the training pipeline. For 6 of our 7 model pairs (Qwen2.5-Coder, Qwen2.5-Math, CodeLlama), neither the SFT data nor the schedule is available to a downstream user; TrajectoryForge needs only the two published checkpoints. When schedule control *is* available, our finding becomes a diagnostic—the

existence of $\alpha^* > 0$ in Fig. 2 shows that cosine decay left performance on the table—orthogonal to that fix rather than in competition with it.

Why not resume fine-tuning for a few more steps? Resuming requires the original training data, unavailable for 6 of 7 pairs. Even when available, a few resumed steps move the model along the *current* gradient at θ_{end}, whereas $\mathbf{\Delta}$ integrates thousands of past gradients weighted by the schedule—a fundamentally different quantity. The direction-task alignment result (IFEval $+1.1\%$ vs. MMLU $\pm 0.1\%$ from the *same* $\mathbf{\Delta}$) shows that $\mathbf{\Delta}$ carries structured task information a local gradient step cannot replicate.

Why not average the last K checkpoints (SWA)? SWA (Izmailov et al., 2018) requires late-stage checkpoints, which released models typically do not distribute. In contrast, TrajectoryForge operates from the two published boundary checkpoints.

Why random search over σ rather than gradient-based optimization. A natural question is whether one could optimize the per-matrix σ coefficients with gradient descent instead of stochastic sampling. This is inapplicable in our target setting: post-hoc enhancement of a *released* model presumes access only to the two published checkpoints and a benchmark, not to the original training data, loss function, or optimizer state. Gradient-based σ optimization would require either (i) re-labeling in-distribution training data—which for 6 of our 7 model pairs is proprietary and unavailable to a downstream user—or (ii) treating benchmark accuracy as a differentiable objective, which it is not. Random per-matrix search side-steps both issues: it needs only forward passes on a small held-out set and is embarrassingly parallel. When the required data *is* accessible, gradient-based refinement of σ is a natural extension we leave to future work.

M A Recipe for Practitioners

The preceding sections establish *when* and *why* TrajectoryForge helps. We close the experimental discussion with an operational summary that a practitioner can run before committing evaluation budget. Given a released base→instruct pair and a target benchmark, we suggest the following three-step protocol.

1. **Pre-hoc feasibility check.** Score θ_{start} and θ_{end} on the target benchmark. If the gap is $\lesssim 1\%$, the direction $\mathbf{\Delta}$ carries little task signal on this task (Section 6.5); the method is expected to be a no-op and search can be skipped. A gap $\gtrsim 3\%$ predicts an actionable regime.
2. **Cheap σ localization.** Run a short pilot with $N = 20$ candidates at three σ values (0.005, 0.05, 0.20) covering the empirically observed range (Appendices F and H). The winning σ of the pilot—together with the model-size trend $\sigma^* \propto 1/\text{model size}$—indicates whether the model lives in the sharp-valley (small σ^*) or flat-plateau (larger σ^*) regime.
3. **Full search and held-out check.** Run the full $N = 200$ sweep concentrated around the pilot winner. To guard against selection bias, evaluate the final candidate on a second, unrelated benchmark of the same modality (e.g. MBPP if HumanEval was used for selection). A candidate that improves on selection and does not degrade on the held-out benchmark is the recommended deployment.

The total cost of this protocol is roughly $N + 60$ inference-only evaluations of a single 7B-scale model—one to two GPU-hours on commodity hardware—and produces a deployable single-model checkpoint with no ensemble overhead.