Replicating Validation-Selected Weight Interpolation on SmolLM2-360M Base and Instruct: ARC, HellaSwag and WinoGrande

Dheiver Francisco Santos, PhD
Independent Researcher
Maceió, Alagoas, Brazil
ORCID: https://orcid.org/0000-0002-8599-9436
Preprint, Figshare, 2026, doi:10.6084/m9.figshare.35014000, CC BY 4.0

Abstract—Selecting an interpolation coefficient between a base language model and its instruct sibling is often presented as a free improvement over both. We replicate a pre-registered test of this claim on SmolLM2-360M and its instruct version, using the same protocol as for Qwen2.5-0.5B with a coarser six-point grid, selection on 300 validation items per task, and 3,967 disjoint test items. The selected $t^* = 0.2$ scores 55.86% on test against 55.40% for the base model: +0.46 pp (95% paired bootstrap CI $[-0.10, +1.02]$, $p = 0.113$). The interval contains zero, so the hypothesis is not supported. Unlike the Qwen pair, the instruct model is worse than the base model on these log-likelihood benchmarks (no chat template), by −5.64 pp (95% CI $[-6.89, -4.35]$). Merge accuracy declines from $t = 0.4$ to 1.0; $t = 0.6$ is not distinguishable from the base. A small-t merge is statistically indistinguishable from the base model (no equivalence test). Two model pairs and one hardware platform limit what can be concluded.

Index Terms—model merging, weight interpolation, task vectors, SmolLM2, replication, negative result

I. Introduction

A single positive result on one model pair is weak evidence that a merging recipe works. Weight interpolation between a base model and its instruction-tuned sibling is a good candidate for replication: it needs no training, it has one free parameter, and the two endpoints are public. A companion experiment with Qwen2.5-0.5B found no detectable advantage of the validation-selected interpolation over the base model. In that pair, however, the instruct and base models were nearly indistinguishable on the benchmarks. Such a pair leaves little room for an interpolated model to win and says little about a pair whose endpoints differ.

This paper repeats the same protocol on SmolLM2-360M [1], with a coarser six-point grid than the eleven points used for Qwen. As we show, the instruct model here is substantially worse than the base model on the log-likelihood benchmarks used. The hypothesis was fixed before any merge was run: an interior t^*, selected on validation items only, has a test macro accuracy that beats the better original model with a paired 95% interval above zero. We make three contributions. First, a second-family replication with the same grid logic, splits, metric and statistics. Second, the full interpolation curve on validation and test, which shows that on this pair the validation split is informative about the harmful region of the path. Third, an explicitly exploratory comparison of the relative size of the weight difference in the two families.

II. Related Work

Replication of merging claims across families. Most merging results are reported on one family at a time. Surveys catalog methods and applications [3], [4], while a study spanning several model sizes reports that merging behavior depends on the base model and its scale [5]. The interpolation idea itself was reported to exceed both endpoints in a vision setting [6]. Whether a selected coefficient transfers across language-model families, with selection and reporting kept apart, is the question we address.

Task vectors. Task arithmetic defines a task vector as the difference between fine-tuned and pretrained weights and edits models by adding scaled vectors [7]; linear interpolation with coefficient t is the same operation with scale t applied to the vector. Work on weight disentanglement argues that linearized fine-tuning in the tangent space makes task vectors compose better [8]. Chat Vector transfers the difference between a chat model and its base into another model [9]. Pruning and sign-resolution schemes reduce interference between several vectors [10], [11]. We use only the single-vector case, and we measure the vector's relative norm as a descriptive quantity.

Merging small models and choosing coefficients. Coefficients can be learned without labels [12], searched by evolutionary optimization [13], or composed from small adapters with gradient-free search [14]. Averaging checkpoints during pretraining has been studied [15], [16], and averaging fine-tuned models from a shared initialization is the idea behind soups [17]. Toolkits implement these operations [18].

The model family. SmolLM2 is a family of small language models trained with a data-centric recipe [1]. The family includes 360M base and instruct checkpoints.

Qwen2.5 provides the contrasting pair used in the companion experiment [2].

Gap. We found no replication that applies one fixed protocol (grid, selection split, test split, paired bootstrap) to base/instruct interpolation in two different small-model families and reports both outcomes, including nulls. This paper provides the second family.

Author's related work. The author's technical notes cover neighboring topics in language-model training: model merging for capability composition [26], knowledge distillation and self-distillation [28], parameter-efficient fine-tuning (LoRA, QLoRA, DoRA) [29], mixture-of-experts training and sparse upcycling [30], curriculum learning and data-efficient pretraining [31], synthetic data generation and self-improvement [32], direct preference optimization [33], RLAIF and Constitutional AI [34], and RLVR and GRPO for reasoning [35]. This paper instead reports a controlled held-out measurement. In a different domain, the author's JEPA preprint [27] uses a similar controlled-comparison style (identical budgets, several seeds, confidence intervals).

III. Method

Let θ_b and θ_i be the parameters of SmolLM2-360M and SmolLM2-360M-Instruct (same architecture and tokenizer), and $\tau = \theta_i - \theta_b$ the task vector. The interpolated model is

$$\theta(t) = (1-t)\,\theta_b + t\,\theta_i = \theta_b + t\,\tau, \qquad t \in [0,1]. \tag{1}$$

All tensors share the same t. The merge is computed in fp32 on CPU and copied into the bf16 scoring model.

Equations and their meaning

Eq. (1) walks from the base model ($t = 0$) to the instruct model ($t = 1$) along a straight line; the second form shows that t scales the instruct update added to the base.

Let $\mathrm{M}^{\mathrm{val}}(t)$ and $\mathrm{M}^{\mathrm{test}}(t)$ be the macro accuracy (mean of the four per-task accuracies) on validation and test. The selected coefficient and the original model to beat are

$$t^* = \arg\max_{t \in \mathcal{G}} \mathrm{M}^{\mathrm{val}}(t), \quad t_o = \arg\max_{t \in \{0,1\}} \mathrm{M}^{\mathrm{val}}(t), \tag{2}$$

with $\mathcal{G} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. In words: choose the best coefficient and the better parent both on validation only. The tested quantity is

$$\Delta = \mathrm{M}^{\mathrm{test}}(t^*) - \mathrm{M}^{\mathrm{test}}(t_o), \tag{3}$$

the paired test gain over the better parent; success means its 95% bootstrap interval lies entirely above zero.

The exploratory descriptor of how large the update is, relative to the model it modifies, is

$$\rho = \frac{\|\theta_i - \theta_b\|_2}{\|\theta_b\|_2}, \tag{4}$$

where both norms run over all parameters concatenated. A larger ρ means the instruct weights moved further from the base weights, relative to the size of the base weights. It does not measure how much the behavior changed.

IV. Experimental Protocol

Models. SmolLM2-360M (base, Hub revision f8027fd0eaee) and SmolLM2-360M-Instruct (a10cc1512eab) [1], Apache-2.0; for the contrast, Qwen2.5-0.5B (060db6499f32) and Qwen2.5-0.5B-Instruct (7ae557604adf) [2]. Full hashes are in MANIFEST.json. Both instruct models are scored with raw prompts, without a chat template; we did not audit their post-training recipes.

Data and splits. ARC-Easy, ARC-Challenge [19], HellaSwag [20] and WinoGrande [21]. ARC uses its official validation and test splits; HellaSwag and WinoGrande use their validation sets, partitioned into disjoint validation and test parts (seed 0). Validation: 300 items per task (299 for ARC-Challenge); test: 1,000 per task (967 for WinoGrande), 3,967 in total. Scoring follows the LM Evaluation Harness conventions [22]: character-normalized log-likelihood for ARC-Easy, ARC-Challenge and HellaSwag; unnormalized partial scoring for WinoGrande.

Selection. All six grid points are scored on validation; t^* and t_o are fixed by Eq. (2) before test is read. To draw the curve, every grid point was also scored on test, and no selection depends on those numbers. Evaluation is deterministic (no training or sampling), so there are no training seeds; randomness is the seed-0 split and the bootstrap seed. Apple Silicon (MPS), bf16, PyTorch 2.8.0, transformers 4.57.6.

Baselines. The primary baseline is the better original by validation, which is the base model ($t_o = 0$, validation 54.03% against 48.61% for instruct). The grid also supplies the control at $t = 1$, the pure instruct model, so every coefficient is scored with the same six-point budget.

Statistics. Accuracy intervals: 95% percentile bootstrap [25] over items, tasks resampled separately and averaged (2,000 resamples). Paired differences: 3,000 resamples, same items for both models, two-sided p from the bootstrap distribution (B=3,000: the smallest nonzero two-sided p is 2/3,000; $p < 0.001$ means no resample crossed zero). The pre-registration (experiments/PREREG.md) fixed 5,000 bootstrap resamples; the analysis used 2,000 for a single accuracy and 3,000 for paired differences. Six primary comparisons were declared across Articles 1–6 (Bonferroni threshold $0.05/6 = 0.0083$). The threshold was added after the pre-registration, whose success rule was a paired 95% CI above zero. Paired item-level intervals are used because eval-set noise is otherwise ignored [23]. This paper's primary comparison is one of them; the instruct-versus-base contrast is descriptive.

V. Results

Table I gives the curve and Fig. 1 plots it.

Primary result. $t^* = 0.2$ scores 55.86% on test against 55.40% for the base model: $\Delta = +0.46$ pp, 95% CI $[-0.10, +1.02]$, $p = 0.113$. The interval contains zero, so the pre-registered hypothesis is not supported; no difference was detected (no equivalence test). Against the

TABLE I

SmolLM2-360M base ($t = 0$) to instruct ($t = 1$) interpolation. Macro accuracy (%) with 95% bootstrap CI on validation and test; paired test difference against the base model (the better original on validation), with 95% CI and nominal two-sided p. Bold: validation-selected t^*. For the primary comparison $p = 0.113$ exceeds both 0.05 and the Bonferroni threshold 0.0083.

t	Validation macro [95% CI]	Test macro [95% CI]	Δ vs. $t = 0$, pp [95% CI]	p
0.0	54.03 [51.27, 56.95]	55.40 [53.89, 56.95]	–	–
0.2	54.20 [51.45, 57.12]	55.86 [54.33, 57.45]	+0.46 [-0.10, +1.02]	0.113
0.4	54.12 [51.45, 57.12]	55.81 [54.33, 57.38]	+0.41 [-0.34, +1.15]	0.282
0.6	53.03 [50.27, 55.87]	54.54 [53.01, 56.14]	-0.86 [-1.79, +0.09]	0.081
0.8	52.20 [49.36, 55.12]	52.48 [51.00, 54.08]	-2.92 [-4.03, -1.81]	<0.001
1.0	48.61 [45.78, 51.53]	49.77 [48.20, 51.39]	-5.64 [-6.89, -4.35]	<0.001

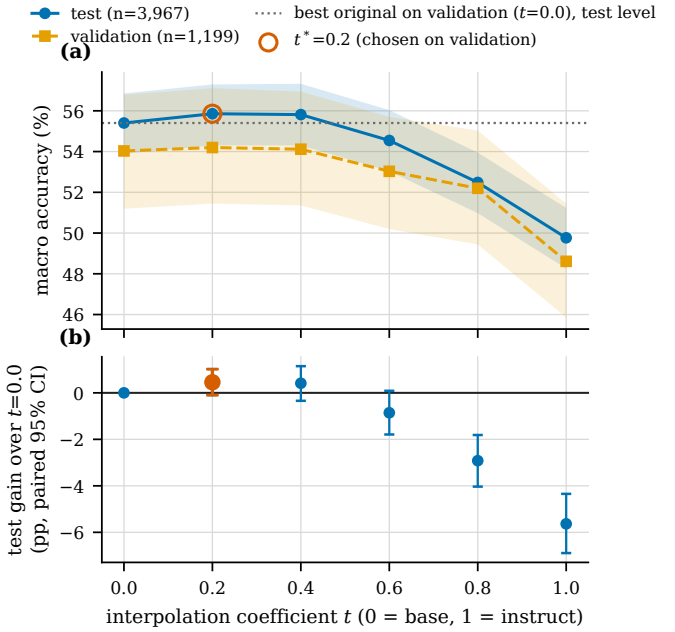


Fig. 1. (a) Macro accuracy along the SmolLM2-360M base-to-instruct path: test (blue) and validation (orange, dashed) with 95% CI bands; red ring marks the validation-selected t^*, dotted line the test level of the better original. (b) Paired test gain over the better original, with 95% CI. Both curves are flat up to $t = 0.4$ and fall beyond it.

instruct model, t^* is better by +6.09 pp ($[+4.83, +7.30]$, $p < 0.001$). The selection margin on validation is small: 54.20% at $t = 0.2$ against 54.12% at $t = 0.4$ and 54.03% at $t = 0$. The point estimate +0.46 pp is close to the Qwen result (+0.65 pp, $[-0.23, +1.52]$) in sign and size. Both intervals include zero, so agreement between the two nulls is not evidence of a shared positive effect. The upper limit of the 95% CI is +1.02 pp over the base model on this test set.

The instruct model is worse than the base model. On test, instruct minus base is −5.64 pp (95% CI $[-6.89, -4.35]$, $p < 0.001$, below the Bonferroni threshold of 0.0083 although this contrast is descriptive); on validation it is −5.42 pp ($[-7.67, -3.08]$). Per task on test (descriptive; no per-task intervals computed), ARC-Easy falls from 69.2% to 50.0%, ARC-Challenge from 38.2% to 34.4%, WinoGrande from 57.8% to 56.6%, while HellaSwag rises from 56.4% to 58.1%. Merged models follow the same trend: $t = 0.6$ is −0.86 pp ($[-1.79, +0.09]$, not distinguishable from zero) and $t = 0.8$ is −2.92 pp ($[-4.03, -1.81]$).

The small-t merge does not inherit the deficit. For $t \leq 0.4$ the paired differences against the base model are +0.46 and +0.41 pp, both intervals containing zero. The merge thus stays statistically indistinguishable from the base model while the endpoint at $t = 1$ is 5.6 pp lower. This describes the path and is not a gain: the selected merge equals the base model within noise.

The validation split is informative here. Validation macro accuracy drops from 54.03% ($t = 0$) to 48.61% ($t = 1$), a range of 5.4 pp. The paired validation difference, instruct minus base, is −5.42 pp ($[-7.67, -3.08]$), so selecting on 300 items per task correctly orders the endpoints. Among $t \leq 0.4$ validation cannot discriminate (spread 0.17 pp), which makes the choice of $t^* = 0.2$ over 0.4 immaterial. This is a contrast with the Qwen pair, where the whole validation curve spans 1.1 pp.

Exploratory contrast with Qwen. The relative task-vector norm of Eq. (4) is $\rho = 0.0794$ for SmolLM2 and 0.0418 for Qwen2.5-0.5B, a factor 1.9. In Qwen, instruct minus base is +0.40 pp on test ($[-0.62, +1.41]$) and −1.00 pp on validation ($[-2.75, +0.92]$), neither distinguishable from zero. This analysis was not pre-registered. The two model pairs differ in family, size, data and post-training recipe, so the data cannot separate ρ from any of these factors, and two points do not define a relationship. We conjecture only that a larger weight difference leaves more room for the endpoints to differ in accuracy. We did not test this, and the result here (a worse instruct model) shows a difference in one direction only.

What cannot be concluded. We cannot conclude that interpolation improves SmolLM2 or fails to improve it by a fraction of a point; that the instruct model is worse in general (the benchmarks are log-likelihood multiple choice without a chat template, which may understate the instruct model); or that ρ predicts merge outcomes.

VI. Discussion and Limitations

In both families the primary hypothesis is null: neither selected merge beats the better parent beyond noise. The second family adds one different fact. Its instruct endpoint is clearly worse, the validation-selected merge lands next to the base model, and large-t merges inherit a loss that the base model lacks. The practical reading is cautious: interpolation toward a worse endpoint is harmful at large t, not detectably harmful at small t, and not shown to be helpful.

Threats to validity. (i) Two model pairs from two families; any cross-family statement rests on $n = 2$. (ii) The benchmarks are four log-likelihood multiple-choice tasks, and the instruct models are scored without chat templates, which may understate them. The per-task drop on ARC-Easy (19 pp) is large, and we did not investigate its cause. (iii) The grid has only six points with step 0.2, coarser than the eleven points used for Qwen, so a narrow optimum would be missed and the selected t^* has less resolution. (iv) The test split is easier than validation (base: 55.4% against 54.0%), so we compare gains, not levels. (v) Possible contamination of benchmarks in either parent. (vi) Selecting among six candidates on one split and reporting on another limits, but does not remove, adaptive-analysis bias [24]. The study also uses a single split (seed 0), one hardware platform, and a bootstrap that treats items as independent. (vii) ρ is a global norm ratio, insensitive to which layers changed. (viii) All evaluations used bfloat16. The fp32 check was run only for Qwen2.5-0.5B, where validation macro accuracy shifted by 0.08 to 0.67 pp; SmolLM2 was not re-evaluated in fp32, so its accuracy levels may shift by a similar amount.

VII. Conclusion

For SmolLM2-360M, validation-selected interpolation between base and instruct ($t^* = 0.2$) is not distinguishable from the base model (no equivalence test): +0.46 pp, 95% CI $[-0.10, +1.02]$ on 3,967 test items. The replication therefore does not support the hypothesis, as in Qwen. The new element is that SmolLM2-Instruct is 5.6 pp below the base model. The small-t merge avoids that deficit but gains nothing detectable beyond the base. The relative norm of the weight difference is nearly twice that of Qwen (0.0794 against 0.0418); we report it as an exploratory contrast, not as an explanation.

Data and Code Availability

Code, per-item caches, tables and figures are in the project repository at git tag preprints-v1 (the commit that contains this manuscript and all scripts and caches). The tables of this paper are regenerated by experiments/art01_06_tables.py and the task-vector norms by experiments/an_taskvec.py (output results/taskvec.json). Datasets: allenai/ai2_arc, Rowan/hellaswag, allenai/winogrande (winogrande_xl). Model revisions are given above and in MANIFEST.json. Versions: Python 3.9.6, PyTorch 2.8.0, transformers 4.57.6, numpy 1.26.4, datasets 4.5.0. The single command ./run_everything.sh regenerates the curves and summaries (models and data are fetched by common/baixar.py). The pre-registration is experiments/PREREG.md, written before any merge was run.

Acknowledgments

No external funding.

References

[1] L. Ben Allal, A. Lozhkov, E. Bakouch, et al., "SmolLM2: when smol goes big—data-centric training of a small language model," arXiv:2502.02737, 2025.

[2] Qwen, A. Yang, et al., "Qwen2.5 technical report," arXiv:2412.15115, 2024.

[3] E. Yang, L. Shen, G. Guo, et al., "Model merging in LLMs, MLLMs, and beyond: methods, theories, applications and opportunities," arXiv:2408.07666, 2024.

[4] W. Li, Y. Peng, M. Zhang, et al., "Deep model fusion: a survey," arXiv:2309.15698, 2023.

[5] P. Yadav, T. Vu, J. Lai, et al., "What matters for model merging at scale?" arXiv:2410.03617, 2024.

[6] M. Wortsman, G. Ilharco, J. W. Kim, et al., "Robust fine-tuning of zero-shot models," in Proc. CVPR, 2022, arXiv:2109.01903.

[7] G. Ilharco, M. T. Ribeiro, M. Wortsman, et al., "Editing models with task arithmetic," in Proc. ICLR, 2023, arXiv:2212.04089.

[8] G. Ortiz-Jimenez, A. Favero, and P. Frossard, "Task arithmetic in the tangent space: improved editing of pre-trained models," in NeurIPS, 2023, arXiv:2305.12827.

[9] S.-C. Huang, P.-Z. Li, Y.-C. Hsu, et al., "Chat Vector: a simple approach to equip LLMs with instruction following and model alignment in new languages," in Proc. ACL, 2024, arXiv:2310.04799.

[10] P. Yadav, D. Tam, L. Choshen, et al., "TIES-Merging: resolving interference when merging models," in NeurIPS, 2023, arXiv:2306.01708.

[11] L. Yu, B. Yu, H. Yu, et al., "Language models are Super Mario: absorbing abilities from homologous models as a free lunch," in Proc. ICML, 2024, arXiv:2311.03099.

[12] E. Yang, Z. Wang, L. Shen, et al., "AdaMerging: adaptive model merging for multi-task learning," in Proc. ICLR, 2024, arXiv:2310.02575.

[13] T. Akiba, M. Shing, Y. Tang, et al., "Evolutionary optimization of model merging recipes," Nat. Mach. Intell., 2025, arXiv:2403.13187.

[14] C. Huang, Q. Liu, B. Y. Lin, et al., "LoraHub: efficient cross-task generalization via dynamic LoRA composition," in Proc. COLM, 2024, arXiv:2307.13269.

[15] S. Sanyal, A. Neerkaje, J. Kaddour, et al., "Early weight averaging meets high learning rates for LLM pre-training," arXiv:2306.03241, 2023.

[16] D. Liu, Z. Wang, B. Wang, et al., "Checkpoint merging via Bayesian optimization in LLM pretraining," arXiv:2403.19390, 2024.

[17] M. Wortsman, G. Ilharco, S. Y. Gadre, et al., "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in Proc. ICML, 2022, arXiv:2203.05482.

[18] C. Goddard, S. Siriwardhana, M. Ehghaghi, et al., "Arcee's MergeKit: a toolkit for merging large language models," arXiv:2403.13257, 2024.

[19] P. Clark, I. Cowhey, O. Etzioni, et al., "Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge," arXiv:1803.05457, 2018.

[20] R. Zellers, A. Holtzman, Y. Bisk, et al., "HellaSwag: can a machine really finish your sentence?" in Proc. ACL, 2019, arXiv:1905.07830.

[21] K. Sakaguchi, R. Le Bras, C. Bhagavatula, et al., "WinoGrande: an adversarial Winograd schema challenge at scale," in AAAI, 2020, arXiv:1907.10641.
[22] L. Gao et al., "EleutherAI/lm-evaluation-harness," Zenodo, doi:10.5281/zenodo.10256836.
[23] E. Miller, "Adding error bars to evals: a statistical approach to language model evaluations," arXiv:2411.00640, 2024.
[24] C. Dwork, V. Feldman, M. Hardt, et al., "Preserving statistical validity in adaptive data analysis," arXiv:1411.2664, 2014.
[25] B. Efron, "Bootstrap methods: another look at the jackknife," Ann. Statist., vol. 7, no. 1, 1979, doi:10.1214/aos/1176344552.
[26] D. F. Santos, "Model Merging for Capability Composition: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610555.
[27] D. F. Santos, "Collapse-Resistant Joint-Embedding Predictive Architectures: A Controlled Comparison of SIGReg, VICReg, and EMA," Zenodo, v2, 2026, doi:10.5281/zenodo.24223364.
[28] D. F. Santos, "Knowledge Distillation and Self-Distillation for Language Models," Zenodo, 2026, doi:10.5281/zenodo.21610100.
[29] D. F. Santos, "Parameter-Efficient Fine-Tuning: LoRA, QLoRA, and DoRA," Zenodo, 2026, doi:10.5281/zenodo.21610033.
[30] D. F. Santos, "Mixture-of-Experts Training and Sparse Upcycling," Zenodo, 2026, doi:10.5281/zenodo.21610062.
[31] D. F. Santos, "Curriculum Learning and Data-Efficient Pretraining: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610595.
[32] D. F. Santos, "Synthetic Data Generation and Self-Improvement: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610525.
[33] D. F. Santos, "Direct Preference Optimization (DPO): A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21609911.
[34] D. F. Santos, "RLAIF and Constitutional AI: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610280.
[35] D. F. Santos, "Reinforcement Learning with Verifiable Rewards (RLVR) and GRPO for Reasoning," Zenodo, 2026, doi:10.5281/zenodo.21609980.