The Language-Modeling Cost Along the Base-to-Instruct Weight Interpolation Path of Qwen2.5-0.5B: WikiText-2 NLL versus 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.35014015, CC BY 4.0

Abstract—Linear interpolation between a base language model and its instruction-tuned version, $\theta(t) = (1-t)\,\theta_{\text{base}} + t\,\theta_{\text{inst}}$, is often tuned for task accuracy alone. We measure what the same path does to language-modeling quality. For Qwen2.5-0.5B and its Instruct version, we evaluate 11 coefficients ($t = 0, 0.1, \ldots, 1$) on a macro-average of four multiple-choice benchmarks (3967 test items) and on WikiText-2 negative log-likelihood (NLL; 100 test blocks). All selection uses a disjoint validation split. The pre-registered hypothesis was that the accuracy-optimal t^* differs from the NLL-optimal one. It holds only literally: validation picks $t^*_{\text{NLL}} = 0.1$ and $t^*_{\text{acc}} = 0.7$, but $t^*_{\text{NLL}} = 0.1$ is not distinguishable from the base (the interval contains zero) and $t^*_{\text{acc}} = 0.7$ rests on about one net validation item (0.0008). Test NLL rises from $t \approx 0.2$ onward, reaching $+0.0886$ nats per token $[+0.0831, +0.0939]$ at $t = 1$. Accuracy shows no detectable difference from the base ($+0.65$ pp, 95% CI $[-0.23, +1.52]$). Evaluation in bfloat16 agrees with fp32 to within 0.003 nats. The study covers one model pair, one scale and one path.

Index Terms—weight interpolation, model merging, instruction tuning, perplexity, Qwen2.5

I. Introduction

Interpolating the weights of a pre-trained model and its fine-tuned descendant is a cheap way to trade one behavior for another. The path has one scalar, t, and every point on it costs one forward pass of evaluation. Practitioners usually select t for the benchmark of interest. Instruction tuning, however, changes the model's output distribution, so the coefficient that is best for multiple-choice accuracy need not be best for modeling raw text.

This paper asks one question: along the base↔instruct path of Qwen2.5-0.5B [1], does the coefficient that maximizes validation accuracy differ from the coefficient that minimizes validation WikiText-2 NLL, and how large is the measured cost of moving between them? A pre-registration fixed the question, the grid and the accuracy metrics before any model was merged (hypothesis 10); for NLL it fixed only the WikiText-2 NLL, and the split into validation blocks 0–39 and test blocks 40–139 is recorded in results/FACTS.md. Selection used only validation items and blocks, and the test split was read once per configuration.

Contributions: (i) a paired, block-level measurement of the NLL cost along the whole path, with the accuracy curve evaluated on the same models; (ii) the observation that the validation-optimal accuracy coefficient is weakly identified, since validation accuracy varies by about one percentage point or less across the path; and (iii) a plain statement of what a correlation over 11 dependent points can and cannot say.

II. Related Work

Interpolating and averaging weights. Interpolation between a zero-shot model and its fine-tuned version improves robustness under distribution shift in WiSE-FT [2], and averaging many fine-tuned models gives model soups [3]. Model Ratatouille recycles diverse fine-tunings of the same pre-trained model to improve out-of-distribution generalization [4]. They target classification accuracy or robustness, not language modeling.

Trade-offs and Pareto fronts. Rewarded soups interpolate weights fine-tuned on different rewards and trace a Pareto front between the rewards [5]. Our setting has two metrics, accuracy and perplexity, on the path between one pair of endpoints; we report both curves but do not analyze a Pareto front. Chat Vector adds the difference between a chat model and its base to another model, extending instruction following across languages [6]; the same difference, scaled by t, is the path measured here. Task arithmetic gives the general view in which such differences are editable directions [7], [8].

When interpolation is well behaved. Interpolation between networks avoids a loss barrier when they are linearly mode connected [9]; TIES resolves interference between task vectors [11], and Fisher weighting weights parameters by their estimated importance [10], and recent work examines merging at scale [12]. Surveys summarize the field [13].

Evaluation. We score multiple-choice tasks by log-likelihood [14] on ARC [15], HellaSwag [16] and WinoGrande [17], and language modeling on WikiText-2 [18]. Statistical reporting follows paired-difference practice [19] and the bootstrap [20]; selecting on one split and reporting on another follows standard practice, although reusing a holdout adaptively can still bias estimates [21]. We did not find a systematic measurement of the raw-text perplexity cost as a function of the interpolation coefficient on a base/instruct pair with block-paired intervals; this paper provides one.

Author's related work. The author's technical notes cover neighboring topics in language-model training: model merging for capability composition [22], direct preference optimization [24], knowledge distillation and self-distillation [25], parameter-efficient fine-tuning (LoRA, QLoRA, DoRA) [26], mixture-of-experts training and sparse upcycling [27], curriculum learning and data-efficient pretraining [28], synthetic data generation and self-improvement [29], and RLAIF and Constitutional AI [30]. This paper instead reports a controlled held-out measurement. In a different domain, the author's JEPA preprint [23] uses a similar controlled-comparison style (identical budgets, several seeds, confidence intervals).

III. Method

Let θ_{base} and θ_{inst} be the weights of Qwen2.5-0.5B and Qwen2.5-0.5B-Instruct (same architecture). For $t \in \{0, 0.1, \ldots, 1\}$,

$$\theta(t) = (1 - t)\,\theta_{\text{base}} + t\,\theta_{\text{inst}}, \tag{1}$$

computed per tensor in fp32 and copied to the model.

Equations and their meaning

Eq. (1) is a straight line in weight space: $t = 0$ is the base model, $t = 1$ is the instruct model, and $t = 0.5$ is the midpoint. Equivalently $\theta(t) = \theta_{\text{base}} + t\,(\theta_{\text{inst}} - \theta_{\text{base}})$, i.e. the instruction-tuning difference scaled by t.

The accuracy metric is the macro-average over the four tasks $\mathcal{T}$,

$$A(t) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \frac{1}{n_\tau} \sum_{i=1}^{n_\tau} y_{\tau i}(t), \tag{2}$$

where $y_{\tau i}(t) \in \{0, 1\}$ is 1 if the model scores the correct answer highest by log-likelihood on item i of task τ. Every task counts equally regardless of its size.

The language-modeling metric is the mean per-block NLL over a set $\mathcal{B}$ of blocks of $L = 512$ tokens,

$$N(t) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \ell_b(t), \quad \ell_b(t) = -\frac{1}{L} \sum_{i=1}^{L} \log p_{\theta(t)}(x_{i+1} \mid x_{\le i}), \tag{3}$$

the average negative log-probability per token, in nats. Lower is better and e^N is perplexity.

The selected coefficients on validation are

$$t^*_{\text{acc}} = \arg\max_t A_{\text{val}}(t), \qquad t^*_{\text{NLL}} = \arg\min_t N_{\text{val}}(t), \tag{4}$$

and the hypothesis is $t^*_{\text{acc}} \neq t^*_{\text{NLL}}$. The cost of a coefficient is the paired difference

$$C(t) = \frac{1}{|\mathcal{B}_{\text{test}}|} \sum_b \big[\ell_b(t) - \ell_b(0)\big], \tag{5}$$

positive when the interpolated model models text worse than the base. Finally, the descriptive association over the 11 grid points is the Pearson correlation $r = \operatorname{corr}\big(A_{\text{test}}(t), N_{\text{test}}(t)\big)$; a positive r means that higher accuracy goes with higher (worse) NLL.

IV. Experimental Protocol

Models. Qwen2.5-0.5B (base) and Qwen2.5-0.5B-Instruct, both Apache-2.0, at Hub revisions 060db649 and 7ae55760 (full hashes in MANIFEST.json) [1]. Evaluation in bfloat16 on Apple Silicon (MPS), PyTorch 2.8.0, transformers 4.57.6; interpolation in fp32 on CPU. There is no training and the evaluation is deterministic; there are no random seeds, and the bootstrap seed is 0.

Data and splits. Accuracy: ARC-Easy, ARC-Challenge, HellaSwag (length-normalized by characters) and WinoGrande (partial scoring, no normalization). Validation uses 300 items per task (299 for ARC-Challenge) and test 1000 per task (967 for WinoGrande), 3967 test items; splits are disjoint (seed 0). ARC uses its official validation and test splits; HellaSwag and WinoGrande items come from their validation sets, partitioned into validation and test. Language modeling: the WikiText-2 test text is tokenized with the Qwen tokenizer and cut into consecutive 512-token blocks; blocks 0–39 are validation and 40–139 are test (100 test blocks).

Selection. Hypothesis 10 (pre-registered): t^*_{acc} differs from t^*_{NLL} along the Qwen base↔instruct path, grid $t = 0, 0.1, \ldots, 1$. Both optima are chosen on validation only (accuracy on 1199 validation items, NLL on 40 blocks), and the test split is used only to report each configuration. The comparison point for gains is the base model ($t = 0$), the best original model on validation accuracy. This is a secondary measurement built on the accuracy grid of the companion LERP experiment (same models, same grid, same items).

Baselines. Base ($t = 0$) and instruct ($t = 1$) are the two original models; the interpolated points are never compared with an oracle chosen on test data.

Statistics. Accuracy: bootstrap over items (2000 resamples for one accuracy, 3000 for paired differences), macro-averaged with tasks equally weighted, 95% percentile CI. NLL: paired bootstrap over test blocks (5000 resamples), 95% CI of $C(t)$. The 100 blocks are consecutive text and not independent draws, so the NLL intervals describe variation across these blocks. The correlation r is computed over 11 points along one path (adjacent points share almost all weights), so it is descriptive and has no inferential interpretation; we give no p for it. The ten NLL comparisons against the base ($t = 0.1, \ldots, 1.0$) are not corrected for multiplicity. At 99.5%

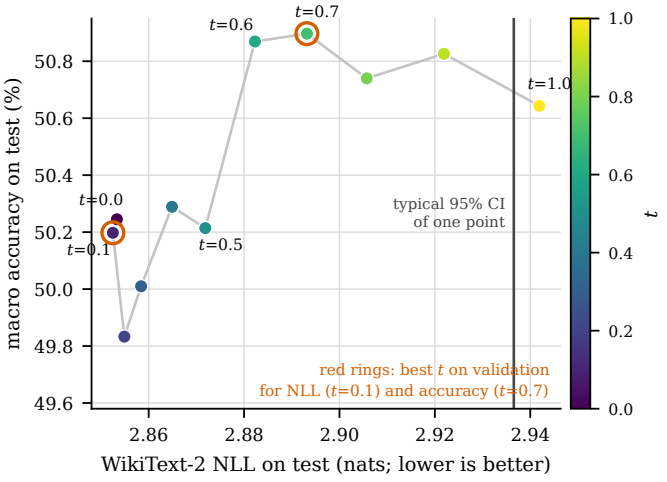


Fig. 1. Test macro accuracy versus test WikiText-2 NLL along the path, one point per t (color). Red rings mark the validation-best t for NLL and for accuracy; the reference bar is a typical 95% CI of one point. NLL rises while accuracy stays within noise. Pearson $r = 0.740$ (descriptive; points are dependent).

(Bonferroni level for ten comparisons), the interval at $t = 0.2$ becomes $[-0.0002, +0.0033]$ and includes zero (results/ci995_art10.txt), so the onset at $t = 0.2$ is not robust; $t = 0.3$ stays above zero. The accuracy comparison against the base at t^*_{acc} has nominal $p = 0.149$. The pre-registration (experiments/PREREG.md) fixed 5,000 resamples. The analysis deviated for accuracy: the paired difference at $t = 0.7$ uses 3000 (analyze.py), and the one at $t = 1$ uses 5000 (experiments/extra_art07_art10.py).

V. Results

Table I lists, for every t, validation and test NLL, the paired NLL cost relative to the base on test blocks, and macro accuracy on validation and test (bootstrap CI over items).

Pre-registered hypothesis. The two selected coefficients differ: $t^*_{\text{NLL}} = 0.1$ and $t^*_{\text{acc}} = 0.7$ on validation. This holds only literally, and the two optima are not equally well determined. Validation NLL at $t = 0$ and $t = 0.1$ differs by 0.0004 nats, and the test difference is -0.0008 $[-0.0018, +0.0001]$, which includes zero. So $t^*_{\text{NLL}} = 0.1$ is not distinguishable from the base. Validation accuracy ranges only from 0.4853 to 0.4961 over the 11 points. There $t^*_{\text{acc}} = 0.7$ (0.4961) beats $t = 0$ (0.4953) by 0.0008, one net item out of 1199 ($1/1200 \approx 0.0008$). A test accuracy has a bootstrap CI of the order of ± 1.5 pp, so the validation argmax is determined largely by noise. The distinct optima therefore reflect a flat accuracy curve and a rising NLL curve rather than two sharply located peaks.

Cost in language modeling. NLL on test blocks is within noise of the base up to $t = 0.1$, then rises monotonically in t in the table: $+0.0185$ $[+0.0162, +0.0209]$ at $t = 0.5$, $+0.0398$ $[+0.0364, +0.0432]$ at $t = 0.7$, and $+0.0886$ $[+0.0831, +0.0939]$ at $t = 1$. All intervals from $t = 0.2$ on lie above zero. In perplexity units, computed from the mean NLL, $t = 0$ gives about 17.3 and $t = 1$ about 19.0 on these 100 blocks, i.e. about 9% higher.

Accuracy along the path. At the validation-selected $t^*_{\text{acc}} = 0.7$ the test accuracy is 0.5090 [0.4938, 0.5246] against 0.5025 for the base: a paired difference of +0.65 pp $[-0.23, +1.52]$, $p = 0.149$, which shows no difference detected (no equivalence test). The instruct endpoint is +0.40 pp $[-0.62, +1.40]$ over the base on the paired macro accuracy. The point estimates are higher at $t \geq 0.6$ than at $t \leq 0.5$ (0.5064–0.5090 versus 0.4983–0.5029), but the intervals overlap throughout, so we do not claim an accuracy gain.

Correlation. Over the 11 points, test accuracy and test NLL have Pearson $r = 0.740$: points with higher accuracy tend to have higher NLL. Because the points lie on one path and adjacent coefficients share nearly the same weights, r does not support a sampling-based statement; it summarizes the shape of the curves in Fig. 1. Its sign only says that the accuracy point estimates and the NLL rise together; it is not a Pareto analysis.

Reading the result. The data give two measurements. The NLL cost is estimated: +0.0398 nats [+0.0364, +0.0432] at $t = 0.7$ and +0.0886 [+0.0831, +0.0939] at $t = 1$. An accuracy gain is not detected: at $t = 0.7$ the upper end of the 95% CI is +1.52 pp, and at $t = 1$ it is +1.40 pp. Gains up to those bounds cannot be excluded, and none is shown.

VI. Discussion and Limitations

Numerical precision. All models were evaluated in bfloat16. We repeated the NLL evaluation in fp32 for all 11 coefficients and the validation accuracy for $t = 0$, 0.7 and 1 (Table II; experiments/an_art10_fp32.py, results/summary_art10_fp32.txt, results/fp32_check_qwen.txt). The bf16−fp32 difference of the test NLL lies in $[-0.0001, +0.0025]$ over the 11 coefficients (validation: at most +0.0026). The fp32 test cost at $t = 1$ is +0.0877 nats, against +0.0886 in bf16. The validation NLL minimum in fp32 is at $t = 0$ (2.8032), not at $t = 0.1$ (2.8037): the two differ by 0.0005 nats, so $t^*_{\text{NLL}} = 0.1$ depends on the precision and is poorly determined; no interval was computed for this gap. Validation macro accuracy in fp32 is 0.4886, 0.4920 and 0.4845 at $t = 0$, 0.7 and 1 (bf16: 0.4953, 0.4961, 0.4853). The order of the three is the same, and the fp32 gap between $t = 0.7$ and $t = 0$ (+0.0034) is of the order of 4 items, and no interval was computed for fp32. The size of the NLL cost and the absence of a detected accuracy gain do not change.

What we conjecture. We conjecture that the NLL cost reflects the shift of the output distribution toward the instruction-tuning data and away from raw web and encyclopedic text; this is an interpretation and was not tested (no probing of the tuning data, no per-token

TABLE I

Qwen2.5-0.5B base↔instruct path. NLL in nats per token on WikiText-2 (validation: 40 blocks; test: 100 blocks); $C(t)$ is the paired test-block difference from $t = 0$ with 95% bootstrap CI. Accuracy: macro-average of four tasks (validation $n = 1199$, test $n = 3967$), 95% bootstrap CI over items.

t	Val NLL	Test NLL	$C(t)$ [95% CI]	Val acc.	Test acc. [95% CI]
0.0	2.8051	2.8533	—	0.4953	0.5025 [0.4875, 0.5174]
0.1	2.8047	2.8525	−0.0008 [−0.0018, +0.0001]	0.4928	0.5020 [0.4863, 0.5171]
0.2	2.8072	2.8549	+0.0016 [+0.0004, +0.0028]	0.4945	0.4983 [0.4831, 0.5135]
0.3	2.8095	2.8584	+0.0051 [+0.0036, +0.0066]	0.4953	0.5001 [0.4844, 0.5150]
0.4	2.8147	2.8649	+0.0115 [+0.0097, +0.0135]	0.4953	0.5029 [0.4874, 0.5183]
0.5	2.8225	2.8719	+0.0185 [+0.0162, +0.0209]	0.4936	0.5021 [0.4869, 0.5174]
0.6	2.8335	2.8823	+0.0289 [+0.0258, +0.0319]	0.4853	0.5087 [0.4940, 0.5241]
0.7	2.8446	2.8932	+0.0398 [+0.0364, +0.0432]	0.4961	0.5090 [0.4938, 0.5246]
0.8	2.8582	2.9057	+0.0524 [+0.0483, +0.0564]	0.4953	0.5074 [0.4920, 0.5226]
0.9	2.8750	2.9219	+0.0686 [+0.0640, +0.0729]	0.4928	0.5083 [0.4929, 0.5241]
1.0	2.8972	2.9419	+0.0886 [+0.0831, +0.0939]	0.4853	0.5064 [0.4909, 0.5220]

TABLE II

Precision sensitivity: mean NLL (nats per token), bf16 versus fp32, at three coefficients, and the validation macro accuracy in bf16 and fp32.

t	Val NLL		Test NLL		Val acc.	
	bf16	fp32	bf16	fp32	bf16	fp32
0.0	2.8051	2.8032	2.8533	2.8517	0.4953	0.4886
0.1	2.8047	2.8037	2.8525	2.8526	—	—
0.7	2.8446	2.8429	2.8932	2.8912	0.4961	0.4920
1.0	2.8972	2.8947	2.9419	2.9394	0.4853	0.4845

analysis, no control with a different tuned model). The measured fact is the cost curve and its size.

Threats to validity. (i) One model pair, one scale (0.5B), one hardware and one path; other tuned models, scales or interpolation operations may differ. (ii) The accuracy effect is within noise, so the result is better read as "NLL cost measured, accuracy benefit not detected" than as a balance between two measured effects. (iii) NLL is measured on one corpus with 100 consecutive blocks; WikiText-2 is encyclopedic text, and the cost may differ on other text. (iv) The validation and test blocks are adjacent stretches of the same corpus. (v) We did not check either model for contamination of the benchmarks or of WikiText-2. (vi) The correlation uses 11 dependent points, and no inference is attached. (vii) We did not evaluate generation quality or instruction following, which an instruction-tuned model is meant to improve. A perplexity cost can be acceptable for a user who wants those. (viii) The tasks are knowledge and common-sense multiple-choice benchmarks scored by log-likelihood, the regime in which a base model is already strong.

Use. A practitioner who selects t by accuracy alone on these benchmarks will not detect a gain and will pay a measurable perplexity cost when t is large. A small t (≤ 0.1) costs nothing detectable in NLL. Whether an interior t is worth its cost depends on a downstream objective that this study does not measure.

VII. Conclusion

Along the Qwen2.5-0.5B base↔instruct interpolation path, the validation-optimal coefficient for accuracy (0.7) differs from that for WikiText-2 NLL (0.1), confirming the pre-registered hypothesis in the literal sense. Test NLL rises with t from about 0.2–0.3 on (the $t = 0.2$ onset is not robust to multiplicity), to +0.0886 nats [+0.0831, +0.0939] at $t = 1$, while accuracy shows no detectable difference from the base (+0.65 pp [−0.23, +1.52] at $t = 0.7$). Test accuracy and NLL correlate at $r = 0.740$ over the 11 dependent points, a descriptive summary of the curve shapes. In fp32 the validation NLL minimum moves to $t = 0$, not distinguishable from $t = 0.1$. The result comes from one pair, one scale and one corpus. Whether an accuracy gain exists, or pays for the cost, is not settled here.

Data and Code Availability

All code, per-item caches and summaries are in the repository artigos-pesos-modelos-pequenos at git tag preprints-v1 (the commit that contains this manuscript and all scripts and caches); the pre-registration is experiments/PREREG.md (written before any merge). Numbers come from results/summary_art10.txt, results/summary_art01.txt (accuracy) and results/extra_art07_art10.txt (per-t NLL differences and the paired accuracy at $t = 1$, script experiments/extra_art07_art10.py) and results/summary_art10_fp32.txt (precision check, script experiments/an_art10_fp32.py). One command regenerates tables and figure: ./run_everything.sh. Versions: Python 3.9.6, PyTorch 2.8.0, transformers 4.57.6, numpy 1.26.4, datasets 4.5.0. Model revisions: Qwen2.5-0.5B 060db6499f32faf8b98477b0a26969ef7d8b9987, Qwen2.5-0.5B-Instruct 7ae557604adf67be50417f59c2c2f167def9a775. Evaluation is deterministic; the bootstrap seed is 0.

Acknowledgments

No external funding.

References

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

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

[3] 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.

[4] A. Ramé, K. Ahuja, J. Zhang, M. Cord, et al., "Model Ratatouille: Recycling diverse models for out-of-distribution generalization," in Proc. ICML, 2023, arXiv:2212.10445.

[5] A. Ramé, G. Couairon, M. Shukor, C. Dancette, et al., "Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards," in Proc. NeurIPS, 2023, arXiv:2306.04488.

[6] S.-C. Huang, P.-Z. Li, Y.-C. Hsu, K.-M. Chen, 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.

[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 Proc. NeurIPS, 2023, arXiv:2305.12827.

[9] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in Proc. ICML, 2020, arXiv:1912.05671.

[10] M. Matena and C. Raffel, "Merging models with Fisher-weighted averaging," in Proc. NeurIPS, 2022, arXiv:2111.09832.

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

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

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

[14] L. Gao et al., "EleutherAI/lm-evaluation-harness," Zenodo, doi:10.5281/zenodo.10256836.

[15] P. Clark, I. Cowhey, O. Etzioni, T. Khot, et al., "Think you have solved question answering? Try ARC, the AI2 reasoning challenge," arXiv:1803.05457, 2018.

[16] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in Proc. ACL, 2019, arXiv:1905.07830.

[17] K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi, "WinoGrande: An adversarial Winograd schema challenge at scale," in Proc. AAAI, 2020, arXiv:1907.10641.

[18] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in Proc. ICLR, 2017, arXiv:1609.07843.

[19] E. Miller, "Adding error bars to evals: A statistical approach to language model evaluations," arXiv:2411.00640, 2024.

[20] B. Efron, "Bootstrap methods: Another look at the jackknife," Ann. Statist., vol. 7, no. 1, pp. 1–26, 1979, doi:10.1214/aos/1176344552.

[21] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, et al., "Preserving statistical validity in adaptive data analysis," in Proc. STOC, 2015, arXiv:1411.2664.

[22] D. F. Santos, "Model Merging for Capability Composition: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610555.

[23] 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.

[24] D. F. Santos, "Direct Preference Optimization (DPO): A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21609911.

[25] D. F. Santos, "Knowledge Distillation and Self-Distillation for Language Models," Zenodo, 2026, doi:10.5281/zenodo.21610100.

[26] D. F. Santos, "Parameter-Efficient Fine-Tuning: LoRA, QLoRA, and DoRA," Zenodo, 2026, doi:10.5281/zenodo.21610033.

[27] D. F. Santos, "Mixture-of-Experts Training and Sparse Upcycling," Zenodo, 2026, doi:10.5281/zenodo.21610062.

[28] D. F. Santos, "Curriculum Learning and Data-Efficient Pretraining: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610595.

[29] D. F. Santos, "Synthetic Data Generation and Self-Improvement: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610525.

[30] D. F. Santos, "RLAIF and Constitutional AI: A Technical Note," Zenodo, 2026, doi:10.5281/zenodo.21610280.