SAME DELTA, DIFFERENT EFFECTS: REUSING POST-TRAINING UPDATES AFTER CONTINUAL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adding a post-training delta after continual pretraining (CPT) can recover instruction following. Does a similar instruction gain mean that the update corrects the same failures at the same cost to language adaptation? We study a fixed delta, defined as the weight difference between a post-trained model and its base, along a Qwen3.5-9B Tibetan CPT trajectory. Overall English instruction gains remain similar, while the language-modeling penalty in bits per byte (BPB) grows roughly fourfold. The corrected prompts also change, even within a fixed set that all receiving checkpoints fail. Scaling reveals a related distinction between preserving scores and preserving individual improvements. At roughly half the BPB cost, a scaled update retains 96% of the full update's net IFEval gain but corrects only 81% of the prompts newly corrected by the full update. Its net IFBench gain retention is 31%. Restricting transfer to lower-half feed-forward blocks shows no clear advantage over scaling at matched cost. An external Llama-3.1-8B/Basque comparison also finds instruction gains with a BPB penalty, although that penalty decreases relative to the base point. These results show why successful reuse requires more than recovering an average instruction score: the same update can change which prompts improve and its cost to language-model fit.

1 INTRODUCTION

Continual pretraining (CPT) adapts a language model to additional text. It is widely used for domain specialization (Xie et al., 2024) and language adaptation (Csaki et al., 2024; Nag et al., 2025), where training resources and model performance vary across languages (Joshi et al., 2020; Ahuja et al., 2023). After CPT, the model must also follow user instructions. Improving its fit to target-language text does not directly train this behavior.

Post-training uses instruction and preference supervision to train such behavior. Instruction tuning can use large task mixtures (Wei et al., 2022; Chung et al., 2024) or small curated datasets (Zhou et al., 2023a). Reinforcement learning from human feedback and direct preference optimization learn from response preferences (Ouyang et al., 2022; Rafailov et al., 2023). Multilingual instruction tuning extends this training across languages (Muennighoff et al., 2023; Shaham et al., 2024). Reusing an existing post-trained model offers an alternative to repeating post-training after each adaptation run.

A post-training delta enables this reuse through parameter arithmetic. The weight difference between a post-trained model and its matched base is added to a checkpoint obtained by CPT. Chat Vector compares this approach with adapting a chat model directly (Huang et al., 2024). DIPLomA examines how the merging coefficient affects instruction following and perplexity (Sarasua et al., 2025), while ParamΔ studies reuse across updated base models (Cao et al., 2025). These studies establish that the update can transfer. We ask how its effects change with the receiving model, both in the prompts it corrects and in the language-adaptation gains it reduces.

Fine-tuning can change access to pretrained capabilities (Kotha et al., 2024), generalization and output diversity (Kirk et al., 2024), and safety alignment (Qi et al., 2024). In model merging, Tao et al. (2024) analyze performance saturation as CPT proceeds. To identify the effect of a transferred update, we must compare the merged model with its own receiving checkpoint. Average accuracy

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.

alone cannot show whether the update corrects the same prompts. Likewise, a smaller language penalty after selective transfer must be compared with simply scaling down the full update.

We hold the delta fixed along a Qwen3.5-9B Tibetan CPT trajectory, tracking instruction gains on English benchmarks and changes in target-language fit and task performance. Paired prompt analysis identifies which failures it corrects. We then vary its scale and parameter subset, matching selective transfer to a full-vector dose at the same BPB cost. Tibetan-prompt subsets assess whether instruction gains extend to target-language inputs. A Post-initialized CPT endpoint and a Llama-3.1-8B/Basque setting provide complementary comparisons.

2 Background and Related Work

Continual pretraining and language adaptation. Continued pretraining on domain- and task-relevant text can improve downstream performance (Gururangan et al., 2020). Latxa adapts Llama models to Basque and provides language-specific evaluation resources (Etxaniz et al., 2024). Corral et al. (2025) study CPT, instruction tuning, and preference alignment in a Basque adaptation pipeline. We examine how reusing a post-training update affects the language gains acquired through CPT.

Reusing post-training updates. Chat Vector adds the difference between matched chat and base models to a continually pretrained model, and compares this approach with continual pretraining of the chat model (Huang et al., 2024). DIPLomA combines target-language CPT, English replay, and delta merging. Its coefficient analysis measures the trade-off between instruction following and perplexity (Sarasua et al., 2025, Section 6.2). ParamΔ studies reuse across updated base models, including CPT models evaluated for domain knowledge and instruction following (Cao et al., 2025).

Tao et al. (2024) report saturation of merged-model performance during CPT and relate it to the loss of SFT parameters during TIES merging. We keep the delta fixed and compare each merged model with its receiver. This separates the update's instruction gains from its language-modeling cost. Prompt-level comparisons further distinguish a similar average gain from correction of the same failures. The matched-cost comparison tests whether choosing a parameter subset improves this trade-off beyond full-vector scaling.

Weight-space composition and selective transfer. Weight averaging and task-vector arithmetic combine or edit model capabilities in parameter space (Wortsman et al., 2022; Ilharco et al., 2023). TIES reduces interference through magnitude trimming and sign resolution (Yadav et al., 2023). DARE randomly drops and rescales parameter differences (Yu et al., 2024), and AdaMerging learns task-wise or layer-wise coefficients (Yang et al., 2024). Tangent-space fine-tuning improves task-vector composition through weight disentanglement (Ortiz-Jimenez et al., 2023). GradFix uses target-model gradient signs to mask transferred task vectors (Rinaldi et al., 2026). We use scaling and parameter selection to study the update's effects, with a matched-cost control for the tested selective update.

3 Study Design and Measurement

We compare a fixed post-training update across CPT checkpoints, then vary its scale and parameter subset. All update effects are measured relative to the receiving checkpoint.

3.1 Models and update transfer

Let B and P denote the parameters of a matched base and post-trained model, and let $\Delta = P - B$ be their difference over the shared parameters of the causal language model. Let C_t denote the base model after processing t CPT tokens, with $C_0 = B$. We evaluate

$$A_{t,\alpha,M} = C_t + \alpha(M \odot \Delta), \tag{1}$$

where α controls update magnitude and M is a binary parameter mask. Full transfer uses $\alpha = 1$ and $M = \mathbf{1}$; no transfer uses $\alpha = 0$. The donor pair is fixed across checkpoints, and injection involves no additional training. Figure 1 summarizes the comparisons.

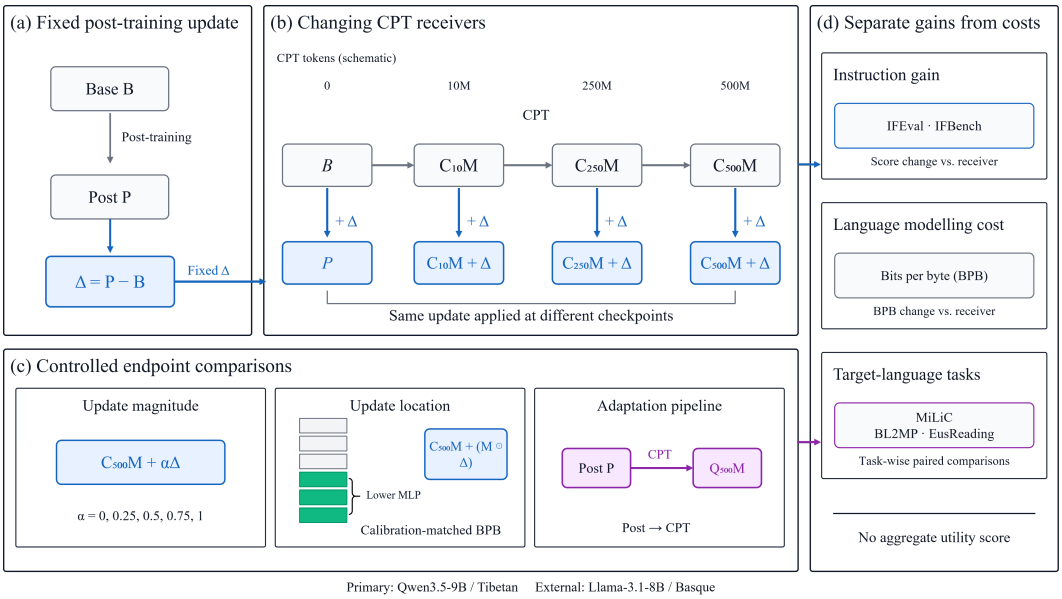

Figure 1: Study overview. (a) The donor update is fixed. (b) Vertically paired models differ only by this update. (c) Endpoint comparisons vary update magnitude, location, or adaptation pipeline. (d) Instruction gains, language-modeling costs, and task changes are measured separately. The Post→CPT branch is a practical pipeline comparison; checkpoint spacing is schematic.

The primary study uses Qwen3.5-9B Base and Post checkpoints (Qwen Team, 2026) and an approximately 500M-token Tibetan CPT trajectory. Training uses native Tibetan sources, drawing on MC^2 (Zhang et al., 2024), with multilingual and general-domain replay. CPT updates all causal-language-model parameters. We evaluate the base point and checkpoints denoted 10M, 250M, and 500M; Appendix A.1 reports their recorded token counts and training configuration.

We also evaluate update reuse in an external Llama-3.1-8B/Basque setting, using the released Llama-eus-8B CPT model and its original Llama Base/Instruct donor pair, alongside the official DIPLomA checkpoint (Sarasua et al., 2025). This setting extends the gain–cost comparison beyond the primary Tibetan trajectory.

3.2 Transferred Gains and Adaptation Costs

For an instruction-following score S_j, we measure the gain from full transfer relative to the receiving model. Language-modeling cost is the corresponding increase in target-language bits per byte (BPB):

$$G_j(t) = S_j(C_t + \Delta) - S_j(C_t), \tag{2}$$

$$K(t) = \mathrm{BPB}(C_t + \Delta) - \mathrm{BPB}(C_t). \tag{3}$$

Positive G_j indicates improved instruction following; positive K indicates worse target-language likelihood. At $t = 0$, these are the original Post–Base differences. Contrasts $G_j(t) - G_j(0)$ and $K(t) - K(0)$ describe how the effect of the same update changes along CPT. Scaled and masked updates use the same receiver-relative definitions. We report target-language task changes separately, without combining heterogeneous scores into a single utility measure. Table 1 summarizes the evaluation dimensions.

Instruction following is assessed with the official rule-based checkers of IFEval (Zhou et al., 2023b) and IFBench (Pyatkin et al., 2025). Strict prompt accuracy requires all constraints to pass. English evaluations share the Post tokenizer and chat template, greedy decoding, and a 1,536-token limit within each family. Qwen thinking is disabled. At the Qwen endpoint, human-translated and proofread Tibetan subsets contain 294 and 79 prompts, retaining required English literals and complete constraint combinations. We use these prompts unchanged with a shared 3,072-token limit and no added response-language requirement. This measures constraint following under Tibetan input, not Tibetan-only response quality (Appendix A.3).

Table 1: Evaluation dimensions and measurement units. English instruction evaluations are shared across model families; Tibetan-prompt subsets supplement the Qwen endpoint comparison.

Dimension	Evaluation set	Reported score
Instruction following	English IFEval / IFBench: 541 / 300 prompts; Tibetan subsets: 294 / 79	Strict prompt accuracy
Language modeling	Tibetan: 5,767 native texts; Basque: 53 reading passages	BPB (lower is better)
Tibetan tasks	MiLiC: 11 task-directions, three ICL sets	Accuracy, chrF++, ROUGE-L
Basque tasks	BL2MP: 1,800 pairs; EusReading: 352 questions	Pair / question accuracy

BPB is total next-token negative log-likelihood in bits divided by the UTF-8 byte count of the evaluated text. Tibetan evaluation uses a fixed native-text test split, separated from calibration and filtered by the recorded lexical overlap audit. Texts are scored with the Base tokenizer in 4,096-token windows. Basque BPB uses the unique EusReading passages, excluding questions and answers. We use BPB for within-language comparisons of language-model fit.

Tibetan task evaluation uses MiLiC-Eval (Zhang et al., 2025) with its Chinese task instructions, raw in-context completion format, and first-line extraction. Nine task types yield eleven directions when article and dialogue translation are evaluated both ways. We average each task's score across three fixed ICL example sets, with identical prompts and task-specific generation budgets across conditions. Basque tasks use BL2MP minimal-pair likelihood preferences and EusReading zero-shot answer-choice likelihoods. These task measures complement BPB rather than serving as interchangeable language-quality estimates.

3.3 Controlled comparisons and the alternative pipeline

At the 9B 500M endpoint, full-vector scaling uses $\alpha \in \{0, .25, .5, .75, 1\}$. Selective transfer examines the lower-half MLP partition (Lower, L; layers 0–15), identified during exploratory development. To compare location with scalar dose, we calibrate a full-vector coefficient to match Lower's BPB on a disjoint native-text calibration split. The coefficient is then frozen for test BPB, instruction following, and MiLiC evaluation. We report the residual test-set BPB difference alongside the matched comparison.

The trajectory analysis follows Base-initialized CPT. Post and Base use the same Tibetan CPT data trajectory. Post→CPT supplies the corresponding 500M-token endpoint reference, denoted $Q_{500\mathrm{M}}$ in Figure 1. Both runs use full-parameter CPT on native Tibetan text with replay; Appendix A.1 gives the training configuration.

3.4 Paired inference and scope

Primary comparisons use 95% percentile intervals from 2,000 paired bootstrap resamples. Instruction-following comparisons resample paired prompts. MiLiC resamples question IDs, retaining all three ICL repetitions and recomputing corpus-level chrF++ in each resample. Basque reading and BPB resample passages; BL2MP resamples sentence pairs. Tibetan BPB is a fixed-corpus estimate. Intervals describe evaluation-sample variation, not training-seed uncertainty, and are unadjusted for multiple comparisons. Supplementary prompt-level analyses use 5,000 resamples; category tests apply multiplicity correction (Appendix B.5).

4 Empirical Characterization of Post-Training Transfer

We begin with full transfer, applying the same Δ to each receiving checkpoint without scaling or masking. This comparison separates changes in the update's effect from changes in the update itself. We examine the Tibetan CPT trajectory, its endpoint task performance, and the external Basque setting.

4.1 Instruction gains and language-modeling costs along CPT

The fixed update improves English instruction following throughout the 9B trajectory (Figure 2). IFEval gains range from 10.2 to 13.7 percentage points (pp), while IFBench gains range from 11.7 to 13.0 pp. At 500M, transfer raises IFEval from 70.6% to 84.3% and IFBench from 21.0% to 34.0%, close to the original Post scores of 84.7% and 34.0%.

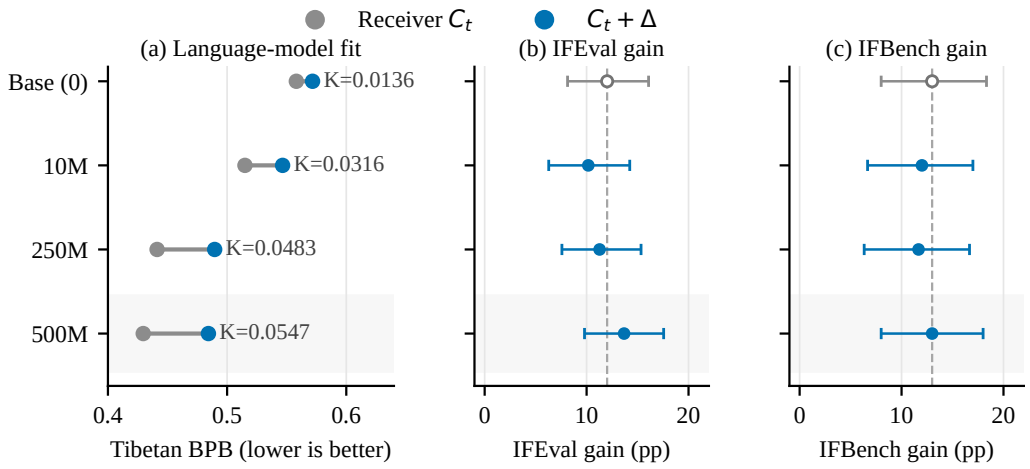

Figure 2: The same update retains English instruction gains while its Tibetan BPB cost increases. (a) Each segment connects the receiver to full transfer; its length is the BPB penalty K. (b–c) Receiver-relative instruction gains with 95% paired-prompt bootstrap intervals. Open circles and dashed lines mark the original Post–Base gains. Rows identify checkpoints, not proportional token distances; BPB is a fixed-corpus estimate.

The BPB penalty increases from 0.0136 at the base point to 0.0316 at 10M tokens, 0.0483 at 250M, and 0.0547 at 500M, roughly four times the original penalty. CPT reduces native Tibetan BPB from 0.5581 to 0.4296; full transfer at 500M raises it to 0.4843. The merged model retains some of the language-modeling improvement, but similar instruction gains now incur a larger language cost.

Similar gains also need not reflect corrections to the same prompts. Consider the fixed set of 219 IFBench prompts that all four receivers fail before injection. Full transfer corrects 43, 45, 41, and 46 of them at the four checkpoints, respectively. Yet 39 prompts (17.8%) change correction status across checkpoints. The receiving model therefore affects which failures the update corrects, not just its BPB cost. The endpoint comparison retains changes in both directions under loose scoring and after excluding capped outputs (Appendix B.5).

4.2 Target-language task effects

The increased BPB cost is accompanied by losses under the fixed MiLiC protocol at 500M. Full transfer lowers scores in eight of eleven task-directions, with all eight paired 95% intervals below zero. Article translation into Tibetan falls from 17.10 to 10.70 chrF++ points, a difference of -6.40 $[-6.76, -6.03]$; reading falls from 78.50% to 73.67%, a difference of -4.83 pp $[-8.33, -1.17]$. Response selection improves by 4.01 pp [1.39, 6.72]; article translation into Chinese and mathematics have smaller positive estimates with intervals spanning zero. Closed-answer losses concern incorrect choices, while translation losses may also involve formatting and termination under raw ICL prompts (Appendix B.4). These are task effects under the stated protocol, not direct measurements of lost linguistic knowledge.

On Tibetan prompts, Full gains 3.06 pp on IFEval $[-3.06, 9.18]$ and 1.27 pp on IFBench $[-7.59, 10.13]$ over CPT. The original Post–Base strict-score intervals also span zero (Table 17), so an established Tibetan-prompt gain is not available as a transfer reference. Full reaches the token limit on 54% and 70% of responses. The wide intervals, especially with 79 IFBench prompts, and frequent capped outputs leave target-language instruction recovery unresolved.

4.3 External comparison on Basque

Full transfer also produces English instruction gains and a BPB cost in the Llama-3.1-8B/Basque setting (Table 2). Relative to the CPT model, IFEval rises from 23.3% to 72.5% and IFBench from 11.7% to 25.3%. BPB increases from 0.8784 to 0.9227, a penalty of 0.0443 [0.0424, 0.0464]. Here, however, this penalty is smaller than the original Post–Base penalty of 0.0617, computed from unrounded values. The two settings share transferable instruction gains accompanied by a language-modeling cost, while the change in that cost depends on the receiving model and adaptation setting.

Table 2: External Basque comparison within the Llama-3.1-8B family. All scores except BPB are percentages; arrows indicate preferred directions. Every row is evaluated under our common protocol, including the released DIPLomA checkpoint (Sarasua et al., 2025).

Model	BPB ↓	IFEval ↑	IFBench ↑	BL2MP ↑	EusReading ↑
Base	1.1862	22.4	15.3	61.3	42.0
Post	1.2480	74.3	24.0	58.7	46.0
CPT	0.8784	23.3	11.7	92.5	40.6
CPT + Δ	0.9227	72.5	25.3	92.2	56.5
DIPLomA	0.9227	71.0	25.7	92.3	55.7

Basque task performance further separates language-modeling fit from downstream utility. Full transfer improves EusReading by 15.9 pp [10.4, 21.5], despite its higher BPB; BL2MP changes by -0.3 pp [-1.4, 0.7]. The released DIPLomA checkpoint yields a closely aligned profile under the same evaluation protocol. Thus, the central distinction is between whether an update still improves instruction following and what language-modeling and task changes accompany those gains. We next examine whether changing the update's magnitude or location improves this relationship.

5 Analysis of Scaling and Selective Transfer

Full transfer restores English instruction-following scores but increases Tibetan BPB. We ask whether reducing the amount of the update or selecting where to apply it can retain those gains at lower language-modeling cost. At the fixed 9B 500M receiving checkpoint, we first vary the full update's dose, then compare selective transfer with scaling at matched calibration cost.

5.1 Different instruction metrics respond differently to scaling

Scaling provides the simplest way to reduce the injected update. As α decreases, the BPB penalty falls, but the accompanying loss of instruction gains differs between benchmarks (Figure 3). At $\alpha = .75$, the penalty is 0.0274 BPB, approximately half that of full transfer. This condition retains 96% of the full update's net IFEval gain but only 31% of its net IFBench gain, with both retention ratios measured relative to the unmodified CPT receiver. Its absolute scores are 83.7% on IFEval and 25.0% on IFBench, compared with 84.3% and 34.0% under full transfer.

The final increase from $\alpha = .75$ to 1 adds only 0.55 pp on IFEval [95% CI: -2.03, 3.14], but 9.00 pp on IFBench [5.00, 13.00]. A coefficient that preserves most of the gain on one benchmark can therefore discard much of the gain on another.

Even on IFEval, retaining 96% of the net gain does not mean retaining the same corrections. Full transfer corrects 99 prompts that CPT fails; $\alpha = .75$ corrects 80 of those, or 81%. Moving from $\alpha = .75$ to full transfer changes 28 prompts from failure to success and 25 from success to failure, leaving a net increase of three passes. Selecting a dose from the nearly flat IFEval curve alone would miss both these changes in successful prompts and the larger loss on IFBench. Appendix B.5 reports the paired counts and constraint-category analysis.

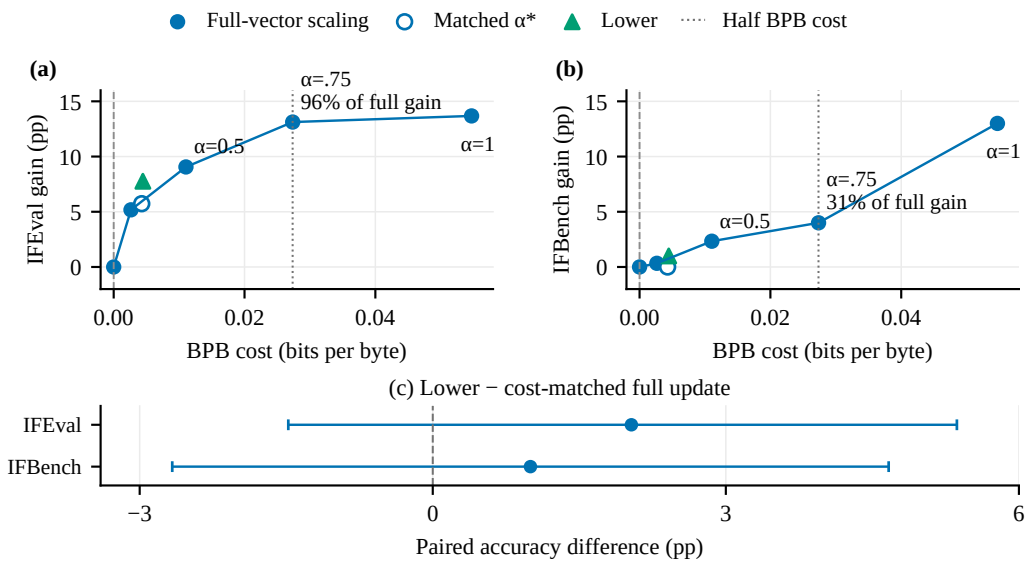

Figure 3: Dose and location at the 9B 500M endpoint. (a–b) Receiver-relative gains and BPB costs; lines connect measured doses, not a fitted frontier. At $\alpha = .75$, roughly half the full BPB cost retains different fractions of the two instruction gains. The open point is the full update with $\alpha^* = 0.315693$, matched to Lower on calibration BPB and plotted at its test cost. (c) Lower minus this matched update, with 95% paired-prompt intervals.

5.2 Selective transfer at matched calibration cost

If instruction benefits and language costs concentrate in different parameter subsets, selective transfer could retain useful updates while excluding those that interfere with adaptation. Lower, which applies the update to the lower-half MLP blocks, provides a concrete test. Its BPB is 0.4341, compared with 0.4843 for full transfer. However, masking changes both the location and the amount of the update. This lower cost alone cannot show whether selecting these parameters improves the trade-off beyond injecting less of the full update.

We therefore compare Lower with a full-vector dose calibrated to the same BPB on a disjoint native-text split. Matching language-modeling cost asks whether parameter selection provides more instruction benefit for a comparable loss of language-model fit. The frozen coefficient is $\alpha^* = 0.315693$; test BPB is 0.434073 for Lower and 0.433897 for the scaled update. The residual is 0.000175, computed before rounding.

At these nearby test costs, Lower's IFEval advantage is 2.03 pp $[-1.48, 5.37]$, and its IFBench advantage is 1.00 pp $[-2.67, 4.67]$. Neither interval excludes zero. MiLiC checks whether the comparison changes for target-language tasks beyond BPB: none of the eleven intervals for Lower minus the matched update lies wholly above zero, while two favor the scaled full-vector update, on title generation and article translation into Tibetan. Thus, the tested partition shows no clear benefit over scaling at matched calibration cost. Its low BPB penalty relative to full transfer does not by itself demonstrate that parameter selection separates instruction benefits from language-adaptation costs.

6 Implications for Adaptation Pipelines

The preceding analyses follow one Tibetan CPT data trajectory under Base and Post initialization, with transfer interventions applied to the Base-initialized endpoint. We compare the resulting endpoints at approximately 500M tokens to determine which combinations of English instruction scores and Tibetan performance the two pipelines provide.

6.1 Adapting Post yields a different endpoint

Post→CPT reaches Tibetan BPB 0.3777, below both Base-initialized CPT (0.4296) and full transfer (0.4843; Table 3). Its English IFEval and IFBench scores fall from Post's 84.7% and 34.0% to 74.1% and 22.7%, whereas full transfer reaches 84.3% and 34.0%. The Post-initialized endpoint thus favors Tibetan fit over the English scores recovered by reuse in these runs.

Table 3: Qwen3.5-9B comparison of Tibetan BPB and strict prompt accuracy (%). Adapted endpoints use approximately 500M CPT tokens. English uses the full benchmarks with a 1,536-token limit; Tibetan uses the 294/79-prompt subsets with a 3,072-token limit. This is not a matched cross-language test.

		English prompts		Tibetan prompts	
Model	BPB ↓	IFEval ↑	IFBench ↑	IFEval ↑	IFBench ↑
Base	0.5581	72.6	21.0	39.1	27.8
Post	0.5717	84.7	34.0	45.6	24.1
CPT	0.4296	70.6	21.0	34.4	25.3
CPT $+ 0.75\Delta$	0.4570	83.7	25.0	37.4	24.1
CPT $+ \Delta$	0.4843	84.3	34.0	37.4	26.6
Post→CPT	0.3777	74.1	22.7	35.7	27.8

6.2 Target-language task gains do not establish instruction retention

Post→CPT's language gains extend to MiLiC (Figure 4). It outperforms full transfer in all eleven directions, with all paired 95% intervals above zero. Against CPT, all eleven point estimates are positive and nine intervals exclude zero; passage classification and title generation are the exceptions. Vocabulary improves by 9.96 pp [7.81, 12.11] and dialogue translation into Tibetan by 11.48 chrF++ points [10.71, 12.23], relative to CPT.

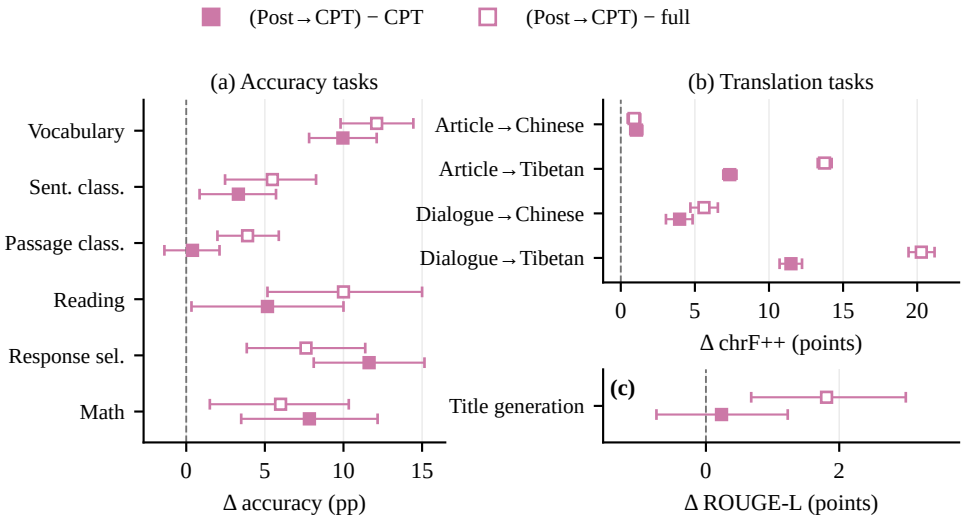

Figure 4: Post→CPT improves MiLiC scores over full transfer in all eleven directions, and over CPT in nine directions with intervals excluding zero. Points show task-wise differences, averaged over three fixed ICL sets; bars are 95% paired bootstrap intervals clustered by question ID. Panels separate (a) accuracy, (b) chrF++, and (c) ROUGE-L, with different horizontal scales. These intervals describe evaluation-sample variation, not training-seed uncertainty.

The Tibetan-prompt comparison remains inconclusive: Post→CPT differs from full transfer by -1.70 pp on IFEval and $+1.27$ pp on IFBench, with both intervals spanning zero. Its outputs reach the 3,072-token limit in 92% and 96% of cases. These bounded-generation scores do not establish equal underlying capabilities; Appendix B.6 reports output diagnostics.

No tested endpoint combines full transfer's English instruction scores with Post→CPT's BPB and MiLiC performance. The practical choice is therefore not resolved by selecting the strongest language model or the strongest English instruction model alone. Reuse recovers English instruction scores; adapting Post yields stronger Tibetan task performance in these runs. Neither result substitutes for checking instruction following in the language of use.

7 Discussion and Conclusion

The same update can have different effects. Along the Tibetan trajectory, similar English instruction gains accompany a growing BPB penalty and changes in which prompts are corrected. The Basque penalty instead decreases relative to its base point. These results support receiver-dependent effects, not universal cost growth. The receiving model determines both the language-modeling cost of reuse and the failures that the update corrects. Average instruction gain therefore captures only one dimension of transfer.

Preserving a score differs from preserving corrections. At $\alpha = .75$, the update retains 96% of the full IFEval net gain but only 81% of the prompts newly corrected by full transfer. The nearly flat aggregate curve hides 28 new passes and 25 new failures between the two doses. IFBench gives a different view of the same intervention: the smaller dose retains only 31% of the full net gain. A single benchmark score can therefore obscure changes in both the identity and type of successful behavior. Evaluating reuse requires aggregate gains, paired prompt outcomes, and more than one instruction benchmark.

Lower cost must be separated from better placement. Restricting the update to lower-half feed-forward blocks reduces its BPB penalty, but it also removes most of the update. Comparing this mask only with full transfer would attribute the effect of a smaller update to parameter location. The calibration-matched full-vector control separates these factors. Lower shows no clear instruction advantage at the same BPB cost, and none of its eleven MiLiC comparisons is clearly positive. This result does not rule out selective transfer; it shows that a location claim requires a dose-matched control.

Adaptation pipelines expose a different trade-off. Applying CPT to Post reaches stronger Tibetan BPB and MiLiC scores than reusing the delta after Base-initialized CPT, while recovering less of the English instruction score. The translated Tibetan constraint sets do not resolve this choice because their intervals span zero and many outputs reach the generation limit. The pipeline comparison therefore adds an endpoint that scaling alone does not produce, rather than identifying one universally preferable order. In practice, pipeline selection depends on whether the deployment prioritizes target-language modeling, target-language tasks, or instruction following in the language of use.

Interpretation and scope. BPB measures language-model fit rather than linguistic competence in full; Basque reading improves despite a BPB penalty. MiLiC supplies task evidence under Chinese-instruction, in-context completion, while the Tibetan-prompt subsets do not establish instruction recovery comparable to the English results. The two model–language settings show that cost need not move in the same direction, but they do not isolate model-family from language effects. The intervals quantify evaluation-sample variation rather than training-seed uncertainty. Crossed model–language settings and matched training replicates would test how broadly the observed receiver dependence holds.

Post-training reuse should be judged by instruction gains and adaptation costs jointly: recovering instruction capabilities does not establish preservation of language adaptation.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting and polishing, and with parts of the experimental code. The authors take responsibility for the final text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

Section 3 defines the transfer operations, comparison groups, measurements, and resampling units. Appendix A specifies model identifiers, CPT training configurations, evaluation protocols, and calibration. Appendix B gives additional trajectory, instruction-following, and task-wise results, including the paired intervals behind the main comparisons.

ETHICS STATEMENT

This study uses existing model checkpoints, text corpora, and benchmark datasets. Automated instruction and language scores do not establish deployment safety or comprehensive competence in Tibetan or Basque. In particular, successful transfer of instruction-following behavior should not be interpreted as preservation of all alignment properties. Use or redistribution of the source models and data must respect their respective licenses and access conditions.

REFERENCES

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4232–4267. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.258. URL https://aclanthology.org/2023.emnlp-main.258/.

Sheng Cao, Mingrui Wu, Karthik Prasad, Yuandong Tian, and Zechun Liu. ParamΔ for direct mixing: Post-train large language model at zero cost. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/78bca5cc621a0846cb1f8265e1927a2a-Abstract-Conference.html.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL https://jmlr.org/papers/v25/23-0870.html.

Ander Corral, Ixak Sarasua Antero, and Xabier Saralegi. Pipeline analysis for developing instruct LLMs in low-resource languages: A case study on Basque. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12636–12655. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.629. URL https://aclanthology.org/2025.naacl-long.629/.

Zoltan Csaki, Bo Li, Jonathan Lingjie Li, Qiantong Xu, Pian Pawakapan, Leon Zhang, Yun Du, Hengyu Zhao, Changran Hu, and Urmish Thakker. SambaLingo: Teaching large language models new languages. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pp. 1–21. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.mrl-1.1. URL https://aclanthology.org/2024.mrl-1.1/.

Julen Etxaniz, Oscar Sainz, Naiara Perez, Itziar Aldabe, German Rigau, Eneko Agirre, Aitor Ormazabal, Mikel Artetxe, and Aitor Soroa. Latxa: An open language model and evaluation suite

for Basque. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14952–14972. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.799. URL https://aclanthology.org/2024.acl-long.799/.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung-yi Lee. Chat vector: A simple approach to equip LLMs with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10943–10959. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.590. URL https://aclanthology.org/2024.acl-long.590/.

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. URL https://openreview.net/forum?id=6tOKwf8-jrj.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6282–6293. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560/.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5a68d05006d5b05dd9463dd9c0219db0-Abstract-Conference.html.

Suhas Kotha, Jacob Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/692ae28fda9bfbde7c01b13bf5a03395-Abstract-Conference.html.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15991–16111. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891/.

Arijit Nag, Soumen Chakrabarti, Animesh Mukherjee, and Niloy Ganguly. Efficient continual pre-training of LLMs for low-resource languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 304–317. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-industry.25. URL https://aclanthology.org/2025.naacl-industry.25/.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems 36*, 2023. doi: 10.52202/075280-2913. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d28077e5ff52034cd35b4aa15320caea-Abstract.html.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, pp. 27730–27744, 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.02833. Datasets and Benchmarks Track.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/83b7da3ed13f06c13ce82235c8eedf35-Abstract-Conference.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Filippo Rinaldi, Aniello Panariello, Giacomo Salici, Fengyuan Liu, Marco Ciccone, Angelo Porrello, and Simone Calderara. Gradient-sign masking for task vector transport across pre-trained models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/59e4f5d009f95324a1a1aa9971b51e25-Abstract-Conference.html.

Ixak Sarasua, Ander Corral, and Xabier Saralegi. DIPLomA: Efficient adaptation of instructed LLMs to low-resource languages via post-training delta merging. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24898–24912. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1355. URL https://aclanthology.org/2025.findings-emnlp.1355/.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. Multilingual instruction tuning with just a pinch of multilinguality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2304–2317. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.136. URL https://aclanthology.org/2024.findings-acl.136/.

Mingxu Tao, Chen Zhang, Quzhe Huang, Tianyao Ma, Songfang Huang, Dongyan Zhao, and Yansong Feng. Unlocking the potential of model merging for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8705–8720. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.508. URL https://aclanthology.org/2024.findings-emnlp.508/.

Jason Wei, Maarten Paul Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew Mingbo Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. Efficient continual pre-training for building domain specific large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10184–10201. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.606. URL https://aclanthology.org/2024.findings-acl.606/.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36*, 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/62868cc2fc1eb5cdf321d05b4b88510c-Paper-Conference.pdf.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 57755–57775. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Chen Zhang, Mingxu Tao, Quzhe Huang, Jiuheng Lin, Zhibin Chen, and Yansong Feng. MC^2: Towards transparent and culturally-aware NLP for minority languages in China. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8832–8850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.479. URL https://aclanthology.org/2024.acl-long.479/.

Chen Zhang, Mingxu Tao, Zhiyuan Liao, and Yansong Feng. MiLiC-Eval: Benchmarking multilingual LLMs for China's minority languages. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11086–11102. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.578. URL https://aclanthology.org/2025.findings-acl.578/.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems 36*, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023b. URL https://arxiv.org/abs/2311.07911.

A Reproducibility Details

A.1 Models and training

Table 4 identifies the released models. The measured Qwen CPT trajectory is shared between Base and Post initializations; the Post-initialized run supplies the corresponding endpoint reference. Updates operate on the causal language model. Lower selects MLP parameters in decoder layers 0–15, and the remainder contains all other shared parameters. Llama uses the same layer-index definition.

Table 4: Released model identifiers. Within each family, the Base and Post rows define the fixed donor pair. The Basque CPT and DIPLomA checkpoints are external releases.

Role	Hugging Face identifier
Qwen Base	`Qwen/Qwen3.5-9B-Base`
Qwen Post	`Qwen/Qwen3.5-9B`
Llama Base	`meta-llama/Llama-3.1-8B`
Llama Post	`meta-llama/Llama-3.1-8B-Instruct`
Basque CPT	`orai-nlp/Llama-eus-8B`
Basque DIPLomA	`orai-nlp/Llama-eus-8B-DIPLomA`

Both Qwen CPT runs update all causal-LM parameters in BF16 with a 4,096-token context, native Tibetan text, and replay. They use $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient clipping at norm 1.0.

CPT configuration. MS-Swift 4.5.2 uses 8-bit AdamW with a custom optimizer plugin, one rank, one sequence per microbatch, and 64 accumulation steps (262,144 packed tokens/step). The cosine schedule has 18,200 steps, with peak learning rate 6×10^{-6} and minimum 8×10^{-7}. Active arguments specify no warmup, consistent with the logged initial rate near 6×10^{-6}, although a separate field records 200 warmup steps. Seed 42 and a 100,000-example shuffle buffer govern streaming interleaving before packing (Table 5). The saved configuration lists Transformers 5.10.2; GPU type, full environment, source-registration code, and custom optimizer implementation are unavailable in the retained records.

Table 5: CPT source aliases and recorded interleaving probabilities, rounded to nine decimals. The common `qwen35_cpt_` prefix is omitted. These are source-sampling weights, not measured token fractions.

Source alias	Probability	Source alias	Probability
`native`	0.777536683	`ufw`	0.005097316
`professional`	0.002437627	`bobaike`	0.004659053
`ocr`	0.000935752	`replay_zh`	0.107291543
`native_web`	0.003181169	`replay_en`	0.062438460
`culturax`	0.001544166	`replay_math`	0.013795538
`l3`	0.013478747	`replay_code`	0.007603946

Table 6 maps nominal Base-CPT budgets to saved trainer states. Each trajectory contrast pairs the same receiver with and without the fixed delta.

Table 6: CPT checkpoints, with processed token counts and learning rates from the saved trainer states.

Nominal budget	Optimizer step	Recorded tokens	Logged learning rate
10M	39	10,153,721	5.999941×10^{-6}
250M	960	249,724,370	5.964384×10^{-6}
500M	1,920	499,116,242	5.858511×10^{-6}

A.2 TIBETAN LIKELIHOOD AND COST CALIBRATION

The native pool contains 8,176 documents. Seed 20260905 assigns 2,048 to calibration and 6,128 to test; lexical-overlap filtering against the available native CPT input removes 119 and 361, leaving 1,929 and 5,767. The audit uses two 64-character anchors followed by exact containment or character-5-gram Jaccard similarity of at least 0.8. Its coverage does not establish exclusion of overlap with the entire externally trained 9B corpus or of semantic overlap.

Documents are tokenized without added special tokens, followed by one EOS, and scored in 4,096-token windows with stride 4,095. BPB divides summed predicted-token negative log-likelihood in bits by original-text UTF-8 bytes, excluding inserted EOS markers from the denominator. All Qwen conditions share the Base-tokenized test stream: 3,442,852 bytes and 1,656,389 predicted tokens.

Let $b(\alpha)$ be full-update calibration BPB and b_L Lower's calibration BPB. From measured endpoints with $b(\alpha_-) \le b_L \le b(\alpha_+)$, calibration proposes $\alpha' = \alpha_- + (b_L - b(\alpha_-))(\alpha_+ - \alpha_-)/(b(\alpha_+) - b(\alpha_-))$. After each measurement, the bracket is updated; a degenerate or repeated proposal falls back to its midpoint. The search allows at most three new measurements and accepts absolute BPB error below 5×10^{-5}. The second proposal, $\alpha^* = 0.3156932466$, gives calibration BPB 0.428481892 versus Lower's 0.428482544. Frozen before testing, it gives test BPB 0.433897198 versus Lower's 0.434072649.

A.3 INSTRUCTION-FOLLOWING EVALUATION

The English IFEval and IFBench sets contain 541 and 300 prompts. All conditions within a family share the Post tokenizer and default chat template, one user message, left padding, batch size 4, greedy decoding, and a 1,536-token limit. Qwen disables thinking and stops on token IDs 248044 or 248046. Llama stops on 128001, 128008, or 128009, with template date fixed to 26 July 2024 and no additional user-supplied system message.

Strict prompt accuracy requires all official constraint checks to pass; loose scoring uses the benchmarks' relaxed extraction rules. Capped responses are retained and scored as produced.

Tibetan-prompt subsets. Human translation and proofreading retain 294 IFEval and 79 IFBench prompts, covering 18 and 19 constraint types. Of the 841 original prompts, 462 were excluded as whole prompts and six deferred for scorer, constraint-consistency, or haiku-metre concerns. The retained sets preserve English literals and all constraints within each prompt; the repeat-prompt arguments in 29 items follow their translated text. We use these prompts unchanged, without adding a Tibetan-response instruction. Generation uses the same Qwen chat and stop settings, but batch size 8 and a 3,072-token limit, identically across all six conditions. The official checkers execute on all retained items. The selected subsets and different token budgets preclude a matched comparison of English and Tibetan scores. Appendix B.6 reports paired effects, loose scores, cap rates, and character-level output diagnostics.

A.4 MILIC TASKS AND SCORING

MiLiC uses Chinese task instructions and raw in-context completion without a chat template (Table 7). Its three supplied example sets are evaluated on the same questions. Non-classification tasks use five examples, except title generation, which uses three. Classification selects examples by a fixed label vocabulary from development and training data, with fixed candidate ordering. Templates truncate classification inputs to 512 characters and title-generation articles to 768. Each example set contains 6,713 task-direction instances, giving 20,139 responses per condition.

Scoring strips whitespace and uses the first output line. Classification uses case-insensitive exact labels; vocabulary, reading, and response selection take an initial A–E answer letter or otherwise compare the extracted string. The mathematics parser takes the last unsigned numeric token, with a missing-number sentinel. Translation uses corpus chrF++ and title generation the benchmark's multilingual ROUGE-L. Scores are averaged across the three fixed example sets.

Table 7: MiLiC evaluation units, metrics, and maximum new tokens. Bo and Zh denote Tibetan and Chinese; translation rows count the same source pairs in both directions. Generation is greedy with batch size 8 and the same Qwen stop tokens as instruction evaluation.

Task-direction	Questions	Token cap	Metric
Vocabulary	900	5	Accuracy
Sentence classification	432	5	Accuracy
Passage classification	504	5	Accuracy
Reading comprehension	200	5	Accuracy
Response selection	407	5	Accuracy
Title generation	900	416	ROUGE-L
Article → Zh	912	100	chrF++
Article → Bo	912	544	chrF++
Dialogue → Zh	673	70	chrF++
Dialogue → Bo	673	192	chrF++
Mathematics	200	150	Accuracy

A.5 Basque evaluation and paired uncertainty

Basque language evaluations use the Llama Base tokenizer for every condition. BPB is computed from the 53 unique EusReading passages, excluding questions and answer options. BL2MP uses 1,800 minimal pairs from `orai-nlp/bl2mp`; a pair is correct when the grammatical sentence has higher summed causal log-likelihood than its alternative. EusReading uses the 352 questions from `HiTZ/EusReading`, with its zero-shot letter-choice prompt and the next-token log-probabilities of the candidate letters. No generated answer text is parsed for these two Basque tasks. The released DIPLomA model is evaluated under this same protocol, rather than importing its published scores.

Paired intervals use 2,000 percentile-bootstrap resamples with random seed 73491. Instruction evaluations resample prompt IDs. MiLiC resamples question IDs, retaining all three example-set repetitions together and using the same sampled IDs across conditions. Corpus chrF++ is recomputed for every resample before averaging across example sets. Basque reading and BPB resample passages, retaining the questions belonging to each sampled passage; BL2MP resamples pairs. No training-seed variation enters these intervals, and no multiple-comparison adjustment is applied.

B Supplementary Measurements

B.1 Trajectory and instruction scores

Tables 8 and 9 give absolute English trajectory scores and strict/loose endpoint scores. Matched uses the frozen coefficient from Appendix A.2.

Table 8: Absolute Qwen3.5-9B trajectory scores. The receiver and receiver plus the fixed full update are evaluated identically. Instruction scores are percentages; at 0 tokens the models are Base and Post.

	Receiver C_t			Receiver $+\Delta$		
CPT tokens	BPB	IFEval	IFBench	BPB	IFEval	IFBench
0	0.558092	72.64	21.00	0.571694	84.66	34.00
10M	0.514969	74.12	22.00	0.546573	84.29	34.00
250M	0.441245	72.09	20.67	0.489561	83.36	32.33
500M	0.429612	70.61	21.00	0.484321	84.29	34.00

Table 9: English instruction scores (%). SP/LP denote strict/loose prompt accuracy. All injected conditions use the 500M CPT receiver.

	IFEval		IFBench	
Condition	SP	LP	SP	LP
Base	72.64	75.05	21.00	24.33
Post	84.66	88.17	34.00	38.33
CPT	70.61	73.57	21.00	25.33
$\alpha = .25$	75.79	78.93	21.33	26.00
$\alpha = .5$	79.67	82.07	23.33	28.33
$\alpha = .75$	83.73	86.51	25.00	29.67
Full	84.29	88.17	34.00	38.67
Lower	78.37	81.89	22.00	27.00
Matched	76.34	79.48	21.00	25.00
Post→CPT	74.12	78.74	22.67	27.33

B.2 Complete task-wise MiLiC comparisons

Table 10 reports the nine conditions with completed MiLiC evaluation. The $\alpha = .75$ condition is evaluated for BPB and instruction following only. Tables 11 and 12 give the paired intervals for the central transfer, matched-cost, and pipeline comparisons. Accuracy differences use percentage points; translation and title-generation differences use their respective metric points. Interval endpoints are rounded for display; whether an interval excludes zero is determined before rounding.

Table 10: MiLiC scores averaged over three fixed ICL sets. Accuracy is a percentage; title generation uses ROUGE-L points and translation uses chrF++ points. Match uses $\alpha^* = 0.3156932466$; P→C denotes Post→CPT. All injected conditions use CPT500M.

Task	Base	Post	CPT	.25	.5	Match	Lower	Full	P→C
Vocabulary	73.30	66.37	76.22	75.85	75.59	75.56	75.26	74.07	86.19
Sentence class.	75.46	74.77	77.78	77.31	77.47	77.16	78.16	75.62	81.10
Passage class.	88.43	86.77	90.61	90.41	89.88	90.28	89.48	87.10	91.01
Reading	67.17	63.67	78.50	77.33	76.67	77.00	76.83	73.67	83.67
Response selection	66.67	58.48	67.57	69.12	70.93	69.53	71.01	71.58	79.20
Title generation	37.83	36.87	40.69	41.56	41.51	41.77	40.99	39.12	40.93
Article → Zh	11.63	11.64	12.82	12.93	13.07	12.87	12.77	12.98	13.88
Article → Bo	12.08	12.16	17.10	15.94	14.61	15.83	15.25	10.70	24.45
Dialogue → Zh	22.48	17.75	26.88	26.78	26.48	26.71	26.51	25.23	30.85
Dialogue → Bo	19.34	18.74	28.42	28.15	26.32	27.60	27.91	19.62	39.90
Mathematics	52.83	46.50	57.83	56.83	58.50	57.83	58.83	59.67	65.67

Table 11: Paired MiLiC effects of full transfer and selective transfer at matched calibration cost. Cells show differences and 95% question-cluster bootstrap intervals. Full minus CPT has intervals below zero in eight directions; Lower minus Match has no interval wholly above zero.

Task	Full − CPT	Lower − Match
Vocabulary	-2.15 $[-3.81, -0.52]$	-0.30 $[-1.04, 0.41]$
Sentence class.	-2.16 $[-4.24, -0.08]$	1.00 $[0.00, 2.08]$
Passage class.	-3.51 $[-5.16, -1.79]$	-0.79 $[-1.65, 0.00]$
Reading	-4.83 $[-8.33, -1.17]$	-0.17 $[-1.67, 1.17]$
Response selection	4.01 $[1.39, 6.72]$	1.47 $[-0.16, 3.11]$
Title generation	-1.58 $[-2.40, -0.75]$	-0.78 $[-1.34, -0.26]$
Article → Zh	0.16 $[-0.13, 0.49]$	-0.10 $[-0.31, 0.11]$
Article → Bo	-6.40 $[-6.76, -6.03]$	-0.57 $[-0.85, -0.29]$
Dialogue → Zh	-1.65 $[-2.33, -1.01]$	-0.20 $[-0.56, 0.14]$
Dialogue → Bo	-8.80 $[-9.51, -8.13]$	0.31 $[-0.11, 0.74]$
Mathematics	1.83 $[-1.50, 5.17]$	1.00 $[-1.33, 3.33]$

Table 12: Paired MiLiC effects of Post→CPT (P→C). Cells show differences and 95% question-cluster bootstrap intervals, using the same saved predictions as Figure 4.

Task	P→C − CPT	P→C − Full
Vocabulary	9.96 [7.81, 12.11]	12.11 [9.81, 14.44]
Sentence class.	3.32 [0.85, 5.71]	5.48 [2.47, 8.26]
Passage class.	0.40 [−1.39, 2.12]	3.90 [1.98, 5.89]
Reading	5.17 [0.33, 10.00]	10.00 [5.17, 15.00]
Response selection	11.63 [8.11, 15.15]	7.62 [3.85, 11.38]
Title generation	0.23 [−0.74, 1.23]	1.81 [0.68, 3.00]
Article → Zh	1.06 [0.64, 1.45]	0.90 [0.46, 1.31]
Article → Bo	7.35 [6.92, 7.81]	13.75 [13.29, 14.23]
Dialogue → Zh	3.97 [3.05, 4.84]	5.61 [4.70, 6.55]
Dialogue → Bo	11.48 [10.71, 12.23]	20.28 [19.42, 21.17]
Mathematics	7.83 [3.50, 12.17]	6.00 [1.50, 10.34]

B.3 Complementary-partition interaction

For disjoint Lower and remainder updates, $\Delta = \Delta_L + \Delta_R$, define the score-level interaction

$$I_j(C) = S_j(C + \Delta_L + \Delta_R) - S_j(C + \Delta_L) - S_j(C + \Delta_R) + S_j(C). \tag{4}$$

Table 13 supports IFBench interaction at both receivers, but not an increase induced by CPT.

The strict IFEval/IFBench scores entering Equation 4 are 78.93%/24.67% for $B+\Delta_L$, 78.19%/24.00% for $B + \Delta_R$, and 75.79%/23.67% for $C_{500\mathrm{M}} + \Delta_R$. Table 9 supplies the remaining constituent scores.

Table 13: Lower/remainder score interactions in percentage points, with 95% paired-prompt intervals. The final row is the difference of the interactions, retaining prompt pairing across all conditions.

Receiving point / contrast	IFEval	IFBench
Base	0.18 [−4.44, 5.18]	6.33 [0.33, 12.67]
CPT500M	0.74 [−3.88, 5.18]	9.33 [3.33, 15.33]
CPT500M − Base	0.55 [−5.36, 6.10]	3.00 [−4.00, 10.00]

B.4 Saved-output diagnostics

MiLiC output. CPT, Full, Lower, Matched, and Post→CPT all produce parseable choice prefixes on vocabulary, reading, and response selection. Full transfer also has no invalid classification labels; its closed-answer losses are incorrect choices rather than format failures. In long-form Tibetan translation, full transfer reaches the cap without a first-line boundary on 2,221/2,736 article outputs and 807/2,019 dialogue outputs, versus 1,425 and 247 for CPT. Scoring complete saved responses still gives losses of −5.60 chrF++ [−5.91, −5.28] and −6.54 [−7.10, −6.02], respectively. First-line extraction alone therefore does not explain the losses, but the comparisons remain conditional on the generation limits. Cap hits do not establish semantic truncation, and BPB is independent of generation termination.

English instruction gains. Table 14 partitions strict-pass gains by whether either paired response is capped. Most net gains occur when neither output is capped. These post-generation strata describe observed outcomes, not a longer-budget counterfactual.

Table 14: Strict-pass differences for full transfer minus the stated reference. Middle cells report the number of paired prompts and the net additional passes, respectively. Contributions sum over all 541 IFEval or 300 IFBench prompts; no condition-specific filtering or rescoring is used. The final column is the full-sample gain in percentage points.

Benchmark	Reference	Neither capped Pairs; net passes	Either capped Pairs; net passes	Gain (pp)
IFEval	CPT	480; +58	61; +16	13.68
IFBench	CPT	213; +30	87; +9	13.00
IFEval	$\alpha = .75$	516; +4	25; −1	0.55
IFBench	$\alpha = .75$	238; +19	62; +8	9.00

B.5 Which prompts does the update correct?

A correction is a prompt that fails before injection and passes afterward; an introduced failure is the reverse. Net gain is their difference divided by the number of evaluated prompts. We pair saved outputs by prompt ID and verify identical prompt text and constraint IDs across conditions. The analyses below describe observed benchmark outcomes, not underlying neural mechanisms. Additional percentile intervals use 5,000 paired-prompt bootstrap samples with seeds 73491 for IFEval and 73492 for IFBench.

Fixed failures across receivers. Table 15 restricts each benchmark to prompts failed by all four receiving checkpoints. A variable correction is a prompt corrected at some, but not all, checkpoints. IFBench has 39 such prompts out of 219 (17.8%; 95% CI: 12.8, 22.8), despite similar total correction counts. IFEval has 18 out of 92 (19.6%; 12.0, 28.3). The common-failure restriction holds the evaluated failures fixed; it is a descriptive subset, not a causal adjustment.

Table 15: Corrections on prompts failed by all four receivers, using strict scoring. Columns Base through 500M count prompts corrected by full transfer. The final column counts distinct prompts whose correction status varies across checkpoints.

Benchmark	Common failures	Base	10M	250M	500M	Variable
IFEval	92	51	49	47	44	18
IFBench	219	43	45	41	46	39

Loose scoring gives 17 variable corrections among 80 common IFEval failures and 37 among 204 IFBench failures. An endpoint check retains prompts failed by both Base and CPT500M and uncapped under Base, Post, CPT500M, and full transfer. On IFEval ($N = 85$), eight are corrected only at Base and five only at CPT500M; on IFBench ($N = 131$), the counts are nine and eight. Loose scoring gives seven and four ($N = 78$), and seven and six ($N = 121$), respectively. Thus, corrections change in both directions beyond capped outputs.

Net gain and shared corrections. Table 16 separates corrections from introduced failures at CPT500M. On IFEval, $\alpha = .75$ retains $71/74 = 95.9\%$ of the full update's net gain [78.8, 116.9], but only $80/99 = 80.8\%$ of its newly corrected prompts [72.9, 88.4]. The former ratio can exceed 100% in bootstrap samples; it is not a proportion of shared corrections. Moving from $\alpha = .75$ to full transfer produces 28 new passes and 25 new failures on IFEval, versus 35 and eight on IFBench. Excluding pairs with either output capped leaves 26 versus 22 on IFEval ($N = 516$), and 26 versus seven on IFBench ($N = 238$). These counts explain how small net changes can coexist with substantial changes in individual outcomes.

Constraint categories. Official ID prefixes define nine IFEval and seven IFBench categories. Exact two-sided paired-prompt swap tests compare full transfer with $\alpha = .75$, retaining constraints within prompts and applying Bonferroni correction across all 16 categories. Only IFBench `format` and `words` pass the 0.05 threshold (adjusted $p = .0146$ and .0293), with constraint-accuracy gains of 14.3 pp [7.1, 22.4] and 13.5 pp [5.7, 21.6]. Together they account for 26 of 34 net additional correct

Table 16: Strict prompt outcomes relative to CPT500M. Each correction/failure cell counts newly correct and newly incorrect prompts, respectively. Shared corrections are prompts corrected by both doses, with the full update's correction count as denominator.

Benchmark	$\alpha = .75$ Corrections / failures	Full Corrections / failures	Net passes .75 / Full	Shared corrections
IFEval	92 / 21	99 / 25	71 / 74	80 / 99
IFBench	31 / 19	53 / 14	12 / 39	24 / 53

constraints. Ten of the 12 net `words` gains come from consonant-counting and palindrome rules, so this is a benchmark-specific pattern, not a general word-level mechanism.

B.6 Tibetan-prompt instruction evaluation

Table 3 gives strict scores for all six conditions. Table 17 reports paired strict effects using 2,000 bootstrap resamples, with seed 73491 reset for each contrast and prompts in dataset order. Loose scoring also leaves the Full minus CPT intervals spanning zero: 2.38 pp [-4.42, 8.84] on IFEval and 1.27 pp [-5.06, 7.59] on IFBench. The original Post minus Base loose difference is 6.12 pp [-0.34, 12.59] on IFEval and -8.86 pp [-16.46, -1.27] on IFBench. The donor pair therefore does not establish a reliable positive Tibetan-prompt instruction gain for transfer to preserve.

Table 17: Tibetan-prompt strict accuracy differences (pp) with paired 95% intervals. All conditions share the translated prompts and 3,072-token limit.

Contrast	IFEval ($N = 294$)	IFBench ($N = 79$)
Post − Base	$6.46\ [-0.34, 12.93]$	$-3.80\ [-13.92, 6.33]$
Full − CPT	$3.06\ [-3.06, 9.18]$	$1.27\ [-7.59, 10.13]$
$\alpha = .75$ − Full	$0.00\ [-5.78, 5.44]$	$-2.53\ [-12.66, 7.59]$
Post→CPT − Full	$-1.70\ [-8.16, 4.42]$	$1.27\ [-8.86, 11.39]$

Table 18: Tibetan-prompt output diagnostics. Loose, capped, and Latin are percentages; Latin is the ASCII-Latin share after removing declared constraint literals. Tokens is the median output length; R_4 is median character-four-gram repetition (%) in capped outputs' final windows, defined below.

	IFEval					IFBench				
Condition	Loose	Capped	Latin	Tokens	R_4	Loose	Capped	Latin	Tokens	R_4
Base	43.88	42.18	12.81	927	71.34	35.44	54.43	24.20	3072	54.15
Post	50.00	39.46	5.88	1107.5	73.72	26.58	62.03	19.07	3072	75.89
CPT	37.07	46.94	9.34	1268.5	70.55	30.38	51.90	10.37	3072	70.75
Full	39.46	54.42	4.34	3072	82.21	31.65	69.62	6.97	3072	85.38
$\alpha = .75$	40.14	47.62	4.73	1410.5	76.88	27.85	60.76	9.52	3072	80.24
Post→CPT	38.10	92.18	5.78	3072	52.57	30.38	96.20	1.48	3072	53.95

For the repetition diagnostic, we collapse whitespace runs to one space and retain punctuation. In the last up to 256 resulting characters, $R_4 = 1 - U_4/N_4$, where U_4 and N_4 count unique and total character four-grams; windows shorter than four characters are excluded. Table 18 reports the median over capped outputs. Post→CPT has higher cap rates but lower local repetition than Full on this measure, so cap frequency alone does not identify repetitive looping. Capping also does not force failure: 88/271 capped Post→CPT IFEval outputs and 20/76 IFBench outputs pass strict scoring.

Among prompts uncapped for both Full and CPT, strict differences are $+12.22$ pp on IFEval ($N = 90$) and $+4.76$ pp on IFBench ($N = 21$). These selected subsets do not estimate a longer-budget outcome. Across conditions, 87.4–98.6% of IFEval responses and 83.5–96.2% of IFBench responses contain Tibetan letters or marks. The Latin share uses counted Tibetan, ASCII-Latin, and Chinese characters after removing declared English literals. Undeclared literals and names can remain. These character measures describe output composition, not semantic quality or response-language correctness.