Online Language Adaptive Sampling for Better Distributed Cross-lingual Gains

Quang Phuoc Nguyen[1*], **Felix Gaschi**[2*], **David Anugraha**[3], **Santiago Martínez Novoa**[4], **En-Shiun Annie Lee**[1,5]
[1]Ontario Tech University [2]Doctrine [3]Stanford University
[4]University of the Andes [5]University of Toronto
quangphuoc.nguyen@ontariotechu.net, felix.gaschi@doctrine.fr,
david.anugraha@stanford.edu, s.martinezn@uniandes.edu.co,
Annie.Lee@ontariotechu.ca

Abstract

Realignment is a promising approach for improving the cross-lingual transfer ability of multilingual language models, particularly for extremely low-resource languages (LRLs). However, existing realignment methods rely on uniform and random sampling of parallel sentences across languages, which may be suboptimal under limited batch sizes. In practice, models may benefit from seeing certain languages more frequently, especially those that are poorly aligned, and the optimal distribution can evolve throughout training. In this work, we propose a simple yet effective adaptive sampling strategy that assigns trainable sampling probabilities to each language. Languages that contribute more to the realignment loss are sampled more frequently in subsequent batches, and the optimal distribution can evolve throughout training. Our method employs an inner–outer optimization loop with a small overhead, leading to consistent performance improvements and, more importantly, distributing the gains across languages. We observed a $+0.67$ average performance increase on all tasks with XLM-R, and $+0.60$ with Gemma 2 9B compared with uniform realignment. Furthermore, our method is robust across different models.[1]

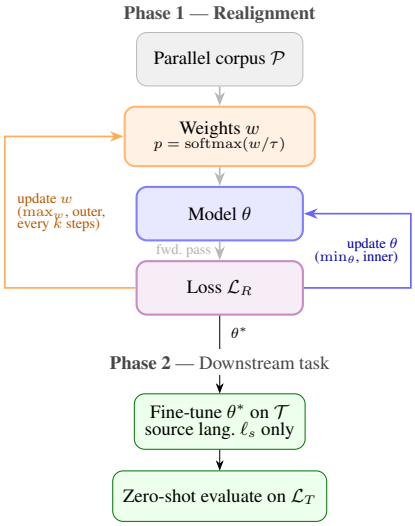


Figure 1: Overview of our method. Phase 1 depicts the realignment process, where the model is trained on parallel data using a contrastive loss. Phase 2 illustrates the downstream cross-lingual transfer.

1 Introduction

Multilingual language models (MLMs) exhibit strong zero-shot cross-lingual transfer, yet still struggle on linguistically distant low-resource languages (LRLs) (Pires et al., 2019). Prior studies attribute much of this failure to weak cross-lingual representation alignment: when language representations are poorly aligned, downstream performance drops sharply (Gaschi et al., 2023; Kargaran et al., 2025). Contrastive realignment on parallel data has emerged as a data-efficient mitigation (Cao et al., 2020; Wu and Dredze, 2020), attractive given the limited data available for LRLs (Anugraha et al., 2025b). Yet its benefits remain inconsistent: some languages improve substantially while others see little gain or even regression (Wu and Dredze, 2020; Efimov et al., 2023).

Prior work on improving realignment has focused mostly on the algorithmic side (Gaschi et al., 2023; Bakos et al., 2025), with little attention to efficient data utilization. Yet strategic data selection improves generalization, efficiency, and robustness across many settings (Wang and Neubig, 2019; Xie et al., 2023; Albalak et al., 2023; Chen et al., 2025; Anugraha et al., 2025a). The closest work, Nguyen et al. (2025), found that the choice of included languages matters, but their selection is static and fixed before training, ignoring that realignment needs may shift as training progresses.

In this work, we propose a dynamic language-

[*]Equal contribution.
[1]Code available at https://github.com/felixgaschi/multilingual-alignment-and-transfer

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 36873–36891
October 24-29, 2026 ©2026 Association for Computational Linguistics

Algorithm 1 Adaptive Sampling for Language-Weighted Realignment

Require: Parallel data $\mathcal{P}$, language pool $\mathcal{L}_T$, inner steps k, temperature τ

Initialize model parameters θ, lang. weights $w \in \mathbb{R}^{|\mathcal{L}_T|}$
for each outer iteration t **do**
  $p \leftarrow \text{softmax}(w/\tau)$ ▷ language sampling distribution
  $L \in \mathbb{R}^{|\mathcal{L}_T|} \leftarrow \mathbf{0}$ ▷ per-language loss accumulator
  for $i = 1, \ldots, k$ **do** ▷ inner loop
    Sample batch $\widetilde{\mathcal{P}}_i$ from $\mathcal{P}$ according to p
    $\theta \leftarrow \theta - \nabla_\theta L_R(\theta; \widetilde{\mathcal{P}}_i)$ ▷ Eq. 1
    Accumulate language contributions to L_R into L
  end for
  $w \leftarrow \text{updateWeights}(L, w)$ ▷ variant-specific
end for

wise sampling strategy for the realignment phase. As shown in Figure 1, rather than sampling languages uniformly, we learn per-language weights from the realignment loss, adaptively oversampling languages that benefit most from alignment (Phase 1), then evaluate via standard zero-shot cross-lingual transfer (Phase 2). Our results show consistent improvements over uniform and static baselines, with gains increasing at scale and distributing more evenly across languages. We further show that the approach generalizes to decoder-only models, extending dynamic sampling beyond encoder-only classification settings.

2 Methodology

In our setup, only the source language (English) carries task supervision; we evaluate zero-shot transfer to a set of target languages $\mathcal{L}_T$. Since target-language labels are unavailable, we use a contrastive realignment loss over parallel data as a language-agnostic proxy that pulls cross-lingual sentence representations together, following Wu and Dredze (2020).

Following Wu and Dredze (2020); Gaschi et al. (2023), for a batch $\mathcal{P}$ of parallel pairs (x, y) of source and target representations,

$$L_R = -\frac{1}{2|\mathcal{P}|} \sum_{(x,y)\in\mathcal{P}} \left[\log \sigma(x, y) + \log \sigma(y, x)\right] \tag{1}$$

where $\sigma(x, y)$ is the in-batch contrastive softmax over cosine similarities with temperature $T_R = 0.1$ (full form in Appendix C.1). The two terms enforce alignment symmetrically.

2.1 Adaptive Language Sampling

We hypothesize that realignment requires more frequent exposure to "difficult" languages (linguistically distant or low-resource), and that these requirements shift as training progresses. Thus, we introduce a bi-level optimization that adaptively samples languages depending on the training progression during realignment. Our framework is summarized in Algorithm 1. We assign a scalar weight w_ℓ to each $\ell \in \mathcal{L}_T$, collected in $w \in \mathbb{R}^{|\mathcal{L}_T|}$, inducing a sampling distribution $p = \text{softmax}(w/\tau)$ with temperature τ. An inner loop optimizes L_R with respect to θ, sampling languages according to p; every k inner steps, an outer loop updates w from the accumulated per-language loss signal so as to *maximize* L_R, focusing sampling on languages that remain hard to align.

UCB-based re-weighting. We frame language selection as a multi-armed bandit, each language an arm whose reward is its per-language realignment loss (Auer et al., 2002). Let n_ℓ be the number of times ℓ has been sampled and $\bar{r}_\ell$ its mean reward; at outer iteration t,

$$\text{UCB}_\ell(t) = \bar{r}_\ell + c\sqrt{\frac{\ln t}{n_\ell + 1}}, \tag{2}$$

where $c > 0$ controls exploration. The first term exploits high-loss (under-aligned) languages; the second explores infrequently sampled ones. Unsampled languages receive a score above the current maximum to guarantee initial coverage. We set $w_\ell \leftarrow \text{UCB}_\ell(t)$.

Gradient-based variant. As an alternative, w can be updated by gradient ascent on the weighted outer-loop loss $\sum_i w_i L_i$, trading UCB's explicit exploration term for a smooth, differentiable update (details in Appendix C.2).

3 Experimental Setup

Models. Our main object of study is **XLM-R Large** (Conneau et al., 2020), a 560M-parameter encoder-only model pretrained on 100 languages. To test generalization to decoder-only models, we additionally include **Gemma 2 9B** (Team et al., 2024), trained with LoRA adapters (Hu et al., 2022) due to resource constraints, following Liu and Niehues (2025). We discuss the rationale for focusing on encoder-only models and classification tasks in Appendix A and justification for using LoRA for generative models in Appendix C.4.

We also perform a post-hoc analysis on models that weren't involved in the other experiments, to demonstrate the off-the-shelf effectiveness of our UCB-based approach (Section 4.5). We include three additional encoder-only models and one decoder-only model: **mBERT** (Devlin et al., 2018), **mDeBERTa v3** (He et al., 2021), **mmBERT** (Marone et al., 2026) and **Llama 3.1 8B** (Grattafiori et al., 2024). These four models constitute the complete set considered for this analysis. For the encoder-only models, we replicate the XLM-R Large training configuration. For Llama 3.1 8B, we mirror the Gemma 2 9B settings by employing LoRA adapters. For the gradient-based experiment, we used the outer-loop learning rate of 10^{-3} for all models.

Data and tasks. We follow Nguyen et al. (2025), evaluating on NER, POS, and NLI across 65 languages (29 low-resource), with parallel realignment data from OPUS-100 and NLLB. Additional dataset details and random seeds are provided in Appendix C.5.

Baselines. We compare against **Fine-tuning only** (no realignment), **Uniform realignment**, and **Most-URIEL** (Littell et al., 2017), the 40 most typologically diverse languages from Nguyen et al. (2025). Our methods are **UCB-based** and **Gradient-based**, all over the same 64 languages. Hyper-parameters were fixed via a small grid search on XLM-R base (Appendix C.3).

4 Results

4.1 Results Overview

Table 1 reports the cross-lingual performance gains of four realignment experiments compared to fine-tuning-only for 2 different model types. All realignment methods improve on the fine-tuning-only baseline for both models, confirming the general effectiveness of realignment. Besides Uniform realignment for POS and gradient-based sampling for NER with Gemma, all realignment methods bring a performance gain ranging from 0.42 to 6.81 points, depending on the method and task.

On XLM-R, both dynamic methods (UCB and gradient-based) outperform the static baselines (Uniform, Most-URIEL). On average, the gradient-based approach achieves the highest gain, at the same time outperforming both uniform sampling and most-URIEL baseline, indicating that the dynamic sampling strategies are more effective than static subset selection or uniform sampling. These gains are consistent over tasks, with the largest improvements observed in NLI, with the UCB-based method achieving a gain of $+1.24$ points over uniform sampling.

On Gemma, the gradient-based method largely fails to generalize, whereas UCB still achieves positive gains across all three downstream tasks, suggesting that the gradient-based approach is more brittle across architectures. Unlike XLM-R, the realignment gains on Gemma are concentrated mainly on NLI tasks, reaching up to 6.81 points of improvement. In some cases, realignment even harms POS tagging and NER performance, although the decoder-only architecture of Gemma is not well-suited for these token-level tasks as it cannot infer a token class from future tokens while encoder-only models can (Dukić and Šnajder, 2024).

All realignment methods achieve significant gains on XNLI for Gemma, at least 6 points, while the gap between the different sampling approaches is smaller than their standard deviation. This suggests that for Gemma, realignment itself is highly effective. Furthermore, the UCB-based method demonstrates strong robustness across architectural changes, performing well not only in the transition from encoder-only to decoder-only models, but also under a different training regime that uses adapters instead of full-model fine-tuning.

4.2 Resource-level Break Down

Table 2 further breaks down models performance across different resource-level groups, including high-resource languages (HRLs), middle-resource languages (MRLs), low-resource languages (LRLs) seen and unseen. The exact language composition of each group is provided in Appendix C.5. The distinction between seen and unseen LRLs is not made for Gemma, as it is not reported which languages were included in its pre-training data.

Compared to static or uniform sampling, dynamic sampling distributes gains more evenly across resource levels for XLM-R. Most importantly, dynamic sampling mitigates the performance drop on MRLs, where uniform sampling triggers a 0.6 point drop over simple fine-tuning, while UCB-based reduces this drop to only 0.3. Furthermore, XLM-R results on unseen languages demonstrate stronger generalization ability compared to the two realignment baselines, suggesting that dynamic sampling leads to more robust and

EXP	NER (F1)	NLI (Acc)	POS (Acc)	AVG
XLM-R-Large				
Fine-tuning only	57.83 $\pm$0.90	65.54 $\pm$0.39	68.90 $\pm$0.61	64.09 $\pm$0.38
Uniform realignment	61.97 $\pm$0.57 (+4.14)	67.70 $\pm$0.34 (+2.16)	71.56 $\pm$0.40 (+2.67)	67.08 $\pm$0.13 (+2.99)
Most-URIEL	62.00 $\pm$0.56 (+4.17)	67.98 $\pm$0.32 (+2.44)	70.93 $\pm$0.35 (+2.03)	66.97 $\pm$0.37 (+2.88)
Gradient-based	**62.25 $\pm$1.10 (+4.42)**	68.65 $\pm$0.29 (+3.11)	**72.33 $\pm$0.34 (+3.44)**	**67.75 $\pm$0.31 (+3.66)**
UCB-based	61.93 $\pm$0.79 (+4.09)	**68.94 $\pm$0.02 (+3.40)**	72.33 $\pm$0.42 (+3.44)	67.73 $\pm$0.13 (+3.64)
Gemma-2-9b				
Fine-tuning only	36.09 $\pm$1.18	63.30 $\pm$3.67	49.09 $\pm$0.94	49.49 $\pm$1.42
Uniform realignment	36.51 $\pm$1.35 (+0.42)	69.97 $\pm$1.08 (+6.67)	48.22 $\pm$4.70 (-0.87)	51.57 $\pm$2.24 (+2.07)
Most-URIEL	**37.60 $\pm$1.17 (+1.51)**	**70.11 $\pm$1.01** (+6.81)	**51.83 $\pm$0.85 (+2.74)**	**53.18 $\pm$0.28 (+3.69)**
Gradient-based	34.54 $\pm$3.94 (-1.55)	69.31 $\pm$0.81 (+6.01)	50.15 $\pm$1.41 (+1.06)	51.33 $\pm$1.45 (+1.84)
UCB-based	37.06 $\pm$0.30 (+0.97)	69.36 $\pm$0.36 (+6.07)	50.06 $\pm$5.97 (+0.97)	52.16 $\pm$1.87 (+2.67)

Table 1: Average performance across three random seeds for each task and across all three tasks, comparing three baselines with two dynamic sampling strategies. All realignment experiments use 16,000 realignment steps.

type	HRL	MRL	LRL_{seen}	$\text{LRL}_{\text{unseen}}$
Fine-tuning only	**75.86**$_{\pm 0.25}$	**77.47**$_{\pm 0.21}$	55.03$_{\pm 0.92}$	43.74$_{\pm 1.01}$
Uniform realignment	75.03$_{\pm 0.62}$	76.87$_{\pm 0.16}$	55.88$_{\pm 0.72}$	53.13$_{\pm 0.10}$
Most-URIEL	74.96$_{\pm 0.08}$	76.95$_{\pm 0.09}$	**56.24**$_{\pm 0.50}$	52.58$_{\pm 0.89}$
Gradient-based	75.14$_{\pm 0.92}$	77.05$_{\pm 0.18}$	55.51$_{\pm 0.75}$	**54.74**$_{\pm 0.27}$
UCB-based	75.02$_{\pm 0.31}$	77.23$_{\pm 0.05}$	55.65$_{\pm 0.21}$	54.54$_{\pm 0.19}$

Table 2: Cross-lingual transfer performance of XLM-R Large in different resource-level language groups.

	XLM-R	Gemma
Avg. gain UCB $-$ Uniform	$+0.67$	$+0.60$
Win rate UCB $>$ Uniform	64.2%	51.0%

Table 3: Average accuracy gap and language-level win rate of UCB-based over uniform realignment (N=441 task–language–seed triplets).

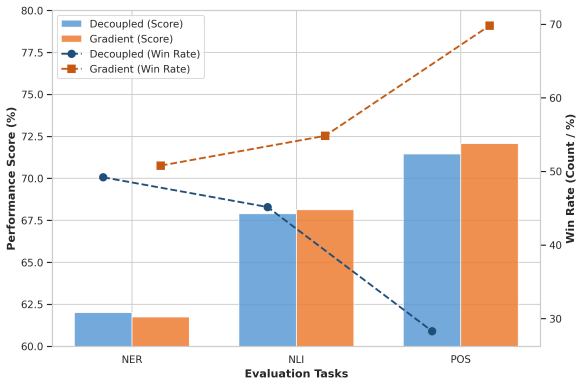


Figure 2: Performance and Win-rate comparison between the gradient-based method and the decoupled variant using seed 42 with 32,000 realignment steps.

generalized cross-lingual transfer. Previous literature has shown that realignment can be harmful to some languages (Wu and Dredze, 2020; Gaschi et al., 2023), our results suggest that this is especially the case for MRLs and HRLs, and that dynamic sampling can mitigate this issue.

4.3 Win Rate vs. Average Gap

The average gains in Table 1 understate the impact of dynamic sampling. Table 3 contrasts the average gap with the language-level win rate, computed across all (task, language, seed) triplets. On XLM-R, UCB only improves the macro-average by $+0.65$ points, but it outperforms uniform realignment on 64.2% of individual triplets, indicating that the gains are spread across the benchmark rather than concentrated on a few languages. On Gemma the effect is smaller (win rate 51.0%). The full pairwise win rate matrix across all methods and both models, together with a detailed analysis, is provided in Appendix D.3.

4.4 Decoupling Language Weights from Dynamic Adaptation

A natural question is whether the gains stem from discovering a better language distribution or from the dynamic co-adaptation of sampling and model parameters. To disentangle these effects, we conduct a decoupling ablation on `XLM-R-Large`: we first learn language weights with the model frozen, then freeze those weights and run realignment using this fixed distribution, followed by standard fine-tuning.

Figure 2 shows that although the downstream task performance differences are relatively small, the gradient-based method achieves better task-wise win–loss rates and more generalized cross-lingual transfer compared to the decoupled variant. This suggests that the benefit of dynamic sampling does not primarily come from converging to a single optimal distribution, but rather from the co-evolution of language weights and model representations throughout training. As realignment reshapes the embedding space, the relative alignment difficulty of each language also changes, re-

	Fine-tuning Only	Uniform	Most URIEL	UCB-based (Ours)	Gradient-based (Ours)
mBERT	56.27 $\pm$0.09	**61.18** $\pm$0.08	60.70 $\pm$0.15	61.17 $\pm$0.25	61.05 $\pm$0.25
mDeBERTa v3	66.36 $\pm$0.30	66.83 $\pm$0.35	66.09 $\pm$0.03	**66.96** $\pm$0.21	66.85 $\pm$0.25
mmBERT	54.72 $\pm$1.30	61.21 $\pm$0.78	61.01 $\pm$0.46	61.48 $\pm$0.76	**61.91** $\pm$0.70
Llama 3.1 8B	42.30 $\pm$0.33	45.10 $\pm$2.92	45.48 $\pm$1.52	45.20 $\pm$2.41	**45.80** $\pm$1.36
XLM-R Large	64.09 $\pm$0.38	67.08 $\pm$0.13	66.97 $\pm$0.37	67.73 $\pm$0.13	**67.75** $\pm$0.31
Gemma-2-9b	49.49 $\pm$1.42	51.57 $\pm$2.24	**53.18** $\pm$0.28	51.33 $\pm$1.87	52.16 $\pm$0.09

Table 4: Macro-average performance of additional encoder-only and decoder-only models on the same three tasks with the same setting and three seeds as XLM-R and Gemma 2. Bold indicates best, and underline indicates second-best in a row. The bottom two rows, separated by the dashed line, are imported from Table 1 for direct comparison. Details per task can be found in Appendix Table 7.

quiring the sampling strategy to adapt dynamically. In particular, language weights that are effective during the early stages of training may become suboptimal later on. As a result, any fixed distribution, even one initially learned from the loss signal, may eventually become misaligned with the model's evolving training needs.

4.5 Robustness across different models

To demonstrate the robustness of our approach across different models, we extend our evaluation to include three additional encoder-only models (mBERT, mDeBERTa v3, and mmBERT) and one decoder-only model (Llama 3.1 8B).

Table 4 indicate that our two proposed approaches outperform both uniform sampling and standard fine-tuning across the majority of evaluated models. While the Most-URIEL baseline surpasses uniform sampling on decoder-only models, it consistently underperforms on encoder-only architectures. Furthermore, although Most-URIEL previously set a strong baseline on Gemma-2-9B, our Gradient-based strategy exceeds it, achieving the highest macro-average results on Llama 3.1 8B, yielding up to a 3.5% point gain over fine-tuning alone. Overall, these results demonstrate that our proposed methods offer superior consistency across diverse model architectures and training recipes over uniform sampling or performing realignment on a fixed subset.

5 Conclusion

Our work introduces two dynamic sampling strategies for parallel data selection during realignment that yield more generalized improvements on `XLM-R-Large`. Furthermore, we extend and validate the benefits of realignment for decoder-only models in multilingual natural language understanding. Lastly, we find that average accuracy understates the advantage of dynamic sampling: gains are more evenly distributed across languages, improving or maintaining LRL results while mitigating the MRL accuracy drop that realignment can trigger. Among our approaches, UCB-based sampling shows greater robustness to training duration and hyperparameter choice, requiring no per-model tuning to remain competitive, whereas gradient-based strategies can achieve larger gains on some architectures (e.g. mmBERT, Llama 3.1 8B) but need model-wise configuration tuning and are more prone to degrading outside their tuned regime. We hope this work encourages further research on data utilization strategies for continual multilingual training and cross-lingual transfer.

Limitations

The conclusions of this work are limited by the scope of our experiments, which focus on encoder-only models and classification tasks, due to a limited availability of resources, both computational and data-related. The compute we had only allowed for realignment with adapters on small LLMs (9B model), and the lack of large-scale multilingual generative benchmarks covering a wide range of LRLs motivated us to focus on classification tasks. Although we include results for a decoder-only model, a more comprehensive evaluation on generative tasks would be a valuable complement to this work.

Due to a lack of available human speakers of the studied low-resource languages, and the large scale of our experiments, we only evaluate on quantitative metrics, which might not fully capture the quality of the realignment and its downstream impact. Future work could focus on a smaller set of languages and include a more qualitative evaluation, especially regarding generative tasks, given the limitations of quantitative metrics for generative tasks.

Because we can only evaluate on the languages

for which we have annotated data in at least one downstream task, our evaluation is limited to 78 languages, which is a small fraction of the world's languages. Although this is a sizable coverage, it still limits the generalizability of our conclusions, and presents the risk of overexposure to certain language families and typological features.

References

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba O. Alabi, Shamsuddeen H. Muhammad, Peter Nabende, Cheikh M. Bamba Dione, Andiswa Bukula, Rooweither Mabuya, Bonaventure F. P. Dossou, Blessing Sibanda, Happy Buzaaba, Jonathan Mukiibi, Godson Kalipe, Derguene Mbaye, and 26 others. 2022. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe Azime, Jian Yun Zhuang, Jesujoba Oluwadara Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, Chiamaka Ijeoma Chukwuneke, Happy Buzaaba, Blessing Kudzaishe Sibanda, Godson Koffi Kalipe, Jonathan Mukiibi, Salomon Kabongo Kabenamualu, Foutse Yuehgoh, Mmasibidi Setaka, Lolwethu Ndolela, and 8 others. 2025. IrokoBench: A new benchmark for African languages in the age of large language models. 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)*, pages 2732–2757, Albuquerque, New Mexico. Association for Computational Linguistics.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. *Preprint*, arXiv:2312.02406.

David Anugraha, Zilu Tang, Lester James V Miranda, Hanyang Zhao, Mohammad Rifqi Farhansyah, Garry Kuwanto, Derry Wijaya, and Genta Indra Winata. 2025a. R3: Robust rubric-agnostic reward models. *arXiv preprint arXiv:2505.13388*.

David Anugraha, Genta Indra Winata, Chenyue Li, Patrick Amadeus Irawan, and En-Shiun Annie Lee. 2025b. ProxyLM: Predicting language model performance on multilingual tasks via proxy models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1981–2011, Albuquerque, New Mexico. Association for Computational Linguistics.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Steve Bakos, David Guzmán, Riddhi More, Kelly Chutong Li, Félix Gaschi, and En-Shiun Annie Lee. 2025. AlignFreeze: Navigating the impact of realignment on the layers of multilingual models across diverse languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 562–586, Albuquerque, New Mexico. Association for Computational Linguistics.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte Miguel Alves, Andre Martins, Ayoub Hammal, Caio Corro, CELINE HUDELOT, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El Haddad, Manuel Faysse, Maxime Peyrard, Nuno M Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. EuroBERT: Scaling multilingual encoders for european languages. In *Second Conference on Language Modeling*.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. *Preprint*, arXiv:2002.03518.

Zhuoyue Chen, Jihai Zhang, Ben Liu, Fangquan Lin, and Wotao Yin. 2025. Scale down to speed up: Dynamic data selection for reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7806–7817, Suzhou, China. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 454–470.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling

human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Cheikh M. Bamba Dione, David Ifeoluwa Adelani, Peter Nabende, Jesujoba Alabi, Thapelo Sindane, Happy Buzaaba, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Perez Ogayo, Anuoluwapo Aremu, Catherine Gitau, Derguene Mbaye, Jonathan Mukiibi, Blessing Sibanda, Bonaventure F. P. Dossou, Andiswa Bukula, Rooweither Mabuya, Allahsera Auguste Tapo, Edwin Munkoh-Buabeng, and 25 others. 2023. MasakhaPOS: Part-of-speech tagging for typologically diverse African languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10883–10900, Toronto, Canada. Association for Computational Linguistics.

David Dukić and Jan Šnajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only LLMs for sequence labeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14168–14181, Bangkok, Thailand. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Annette Rios, Gustavo A. Giménez-Lugo, Ivan Vladimir Meza Ruiz, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Ngoc Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6351–6377. Association for Computational Linguistics.

Pavel Efimov, Leonid Boytsov, Elena Arslanova, and Pavel Braslavski. 2023. *The Impact of Cross-Lingual Adjustment of Contextual Word Representations on Zero-Shot Transfer*, page 51–67. Springer Nature Switzerland.

Ahmed Elshabrawy, Thanh-Nhi Nguyen, Yeeun Kang, Lihan Feng, Annant Jain, Faadil Abdullah Shaikh, Jonibek Mansurov, Mohamed Fazli Mohamed Imam, Jesus-German Ortiz-Barajas, Rendi Chevi, and Alham Fikri Aji. 2025. Statement-tuning enables efficient cross-lingual generalization in encoder-only models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16226–16248, Vienna, Austria. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Felix Gaschi, Patricio Cerda, Parisa Rastin, and Yannick Toussaint. 2023. Exploring the relationship between alignment and cross-lingual transfer in multilingual transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3020–3042, Toronto, Canada. Association for Computational Linguistics.

Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Aung Kyaw Htet and Mark Dras. 2025. Myanmar xnli: Building a dataset and exploring low-resource approaches to natural language inference with myanmar. *Preprint*, arXiv:2504.09645.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. 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*, pages 6282–6293, Online. Association for Computational Linguistics.

Amir Hossein Kargaran, Ali Modarressi, Nafiseh Nikeghbal, Jana Diesner, François Yvon, and Hinrich Schuetze. 2025. MEXA: Multilingual evaluation of English-centric LLMs via cross-lingual alignment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 27001–27023, Vienna, Austria. Association for Computational Linguistics.

Patrick Lewis, Barış Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL

and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Danni Liu and Jan Niehues. 2025. Middle-layer representation alignment for cross-lingual transfer in fine-tuned LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15979–15996, Vienna, Austria. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. In *Transactions of the Association for Computational Linguistics*, volume 9, pages 1389–1406.

Rahmad Mahendra, Alham Fikri Aji, Samuel Louvan, Fahrurrozi Rahman, and Clara Vania. 2021. IndoNLI: A natural language inference dataset for Indonesian. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10511–10527, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2026. mmBERT: A modern multilingual encoder with annealed language learning. In *Forty-third International Conference on Machine Learning*.

Quang Phuoc Nguyen, David Anugraha, Félix Gaschi, Jun Bin Cheng, and En-Shiun Annie Lee. 2025. Rethinking what matters: Effective and robust multilingual realignment for low-resource languages. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1877–1905, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *Preprint*, arXiv:1803.02999.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2019. Learning to reweight examples for robust deep learning. *Preprint*, arXiv:1803.09050.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanna Shcharbakova, Tatiana Anikina, Natalia Skachkova, and Josef Van Genabith. 2025. When scale meets diversity: Evaluating language models on fine-grained multilingual claim verification. In *Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER)*, pages 69–84, Vienna, Austria. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Jörg Tiedemann. 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248. University of Helsinki.

Xinyi Wang and Graham Neubig. 2019. Target conditioned sampling: Optimizing data selection for multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5823–5828.

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. *Preprint*, arXiv:2305.10429.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020a. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020b. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

A Scope: classification tasks and choice of models

We focus on classification rather than generative tasks because existing multilingual generative benchmarks have limited language coverage. Question Answering datasets such as TyDiQA, MLQA, and MKQA typically cover at most 11 languages (Clark et al., 2020; Lewis et al., 2020; Longpre et al., 2021), far below the 65-language coverage of the classification benchmarks we use. A comprehensive evaluation on generative tasks is left to future work.

For classification, encoder-only architectures remain preferred due to their size and efficiency (Dukić and Šnajder, 2024; Shcharbakova et al., 2025; Elshabrawy et al., 2025), motivating continued investment in recent encoder-only models such as EuroBERT (Boizard et al., 2025) and mmBERT (Marone et al., 2026). This motivates our main focus on XLM-R Large. To verify that our adaptive sampling approach generalizes beyond encoder-only models, we additionally evaluate it on Gemma 2 9B as a decoder-only test point, following Liu and Niehues (2025).

B Related Works

Cross-lingual realignment. Bitext-based contrastive realignment pulls together cross-lingual sentence representations by applying a symmetric contrastive loss over in-batch negative pairs sampled from a parallel corpus (Cao et al., 2020; Wu and Dredze, 2020). This approach requires no task-specific supervision, but gains vary substantially across languages and tasks (Wu and Dredze, 2020). Gaschi et al. (2023) hypothesized that catastrophic forgetting is a systematic failure mode in encoder-only models undergoing realignment; Bakos et al. (2025) mitigated this with AlignFreeze, which selectively freezes model components during training. All four works evaluated on a limited set of predominantly high-resource or European languages.

The emergence of large-scale LRL benchmarks - driven by the Masakhane initiative (Adelani et al., 2022; Dione et al., 2023; Adelani et al., 2025) - made it newly tractable to assess realignment across typologically diverse, low-resource languages. Nguyen et al. (2025) aggregated those low-resource datasets with XTREME-R (Hu et al., 2020), performing realignment in 65 languages including 29 LRLs and evaluating on three tasks and two models. Their key finding is that realignment is particularly impactful for LRLs — especially those unseen during pre-training, where improvements reach up to 10 points — while offering limited returns for high-resource languages. They also show that a linguistically diverse language subset can match or outperform the full pool.

Our work builds directly on this line of research. Nguyen et al. (2025) suggest that some languages might benefit more from realignment than others, but their approach is static and binary: decided once before training begins, with no account of how alignment difficulty shifts across languages as training progresses. We replace this with continuous, online per-language weights learned during realignment itself.

Dynamic language sampling. Dynamic data selection methods have been developed to better allocate training budget based on the informativeness of samples at a given stage of learning (Chen et al., 2025). At the language level, multilingual pretraining research has studied how to adjust the mixture across languages to improve coverage of low-resource ones (Xie et al., 2023; Albalak et al., 2023).

However, dynamic sampling within the realignment phase remains underexplored. Existing realignment methods rely on static, uniform sampling of parallel corpora, which ignores how alignment difficulty varies across language pairs and evolves throughout training. Our work addresses this gap by introducing an online, loss-driven sampling strategy that adapts per-language weights during realignment, without requiring external supervision or a held-out validation set.

Bilevel optimization. Bilevel optimization uses a hierarchical inner–outer loop structure, popularized by MAML (Finn et al., 2017) for meta-learning model initialization and extended to first-order approximations by Reptile (Nichol et al., 2018) for scalability. Ren et al. (2019) adapted this framework to dynamic data reweighting, assigning per-example weights based on their gradient alignment with a clean, held-out validation set.

While our inner-outer loop structure is inspired by this line of work, our setting differs in key ways, and doesn't constitute a meta-learning setup. We do not have access to a clean validation set with downstream labels for target languages, so our outer loop optimizes language weights based on the realignment loss itself, and can be seen as a form of online curriculum learning that prioritizes hard-to-align

languages, rather than a meta-learning approach that optimizes for generalization to unseen tasks.

C Details about the methodology

C.1 Full form of the realignment loss

The contrastive term $\sigma(x, y)$ used in Equation 1 is

$$\sigma(x,y) = \frac{\exp\left(\frac{\text{sim}(x,y)}{T_R}\right)}{\sum_{h\in\mathcal{H},\, h\neq x} \exp\left(\frac{\text{sim}(x,h)}{T_R}\right)}, \quad (3)$$

where sim is cosine similarity, $T_R = 0.1$ (Wu and Dredze, 2020), and $\mathcal{H}$ is the set of all source and target representations in the batch. The symmetric pair $\log \sigma(x, y) + \log \sigma(y, x)$ enforces that x is closer to y than to any other element in $\mathcal{H}$, and vice versa.

C.2 Gradient-based variant

For the gradient-based variant, the outer-loop loss is the weighted sum of the accumulated per-language realignment losses over the k inner steps:

$$L_{\text{outer}} = \sum_{i=1}^{|\mathcal{L}_T|} w_i L_i. \quad (4)$$

This loss has a closed-form maximizer that puts all the mass on the language with the highest loss; instead we update w by a single gradient-ascent step, producing a smoother trajectory. Compared to UCB, this variant forgoes explicit exploration but produces a smooth, differentiable update signal. The connection between the two variants is analyzed in Appendix C.6.

C.3 Hyper-parameters

Hyper-parameters were fixed via a small grid search on XLM-R base. For the UCB-based method, we set the exploration coefficient c=0.1 and the number of inner steps k=5. For the gradient-based method, we use k=10 inner steps and an outer-loop learning rate of 10^{-3}, increased to 10^{-2} for Gemma to compensate for the LoRA adapters' smaller gradient magnitudes.

During realignment, for XLM-R and Gemma 2, we used (respectively) a batch size of 128 and 16, a learning rate of 7.5×10^{-6} and 2×10^{-2}, with 16k realignment steps unless mentioned otherwise.

During fine-tuning, for XLM-R and Gemma 2, we used (respectively) a batch size of 128 and 32, the same learning rate as realignment, and during 5 epochs for each task, except NLI which is trained for 2 epochs due to its significantly larger size, following (Nguyen et al., 2025).

For LORA adapters with Gemma, we use rank $r = 8$, scaling factor $\alpha = 32$, and a LoRA dropout of 0.1, with the adapter targeting the PEFT library's default set of attention projection modules. The same adapter is shared between the realignment and fine-tuning phases; all non-adapter parameters of the backbone are kept frozen.

C.4 On LoRA Usage in the Decoder-Only Setting

EXP	NLI Score	PoS Score	AVG
Uniform realignment (w/o LoRA)	42.47	31.42	36.94
Uniform realignment (LoRA)	**61.43**	**45.68**	**53.56**

Table 5: Ablation study on the Gemma-2-2B model comparing uniform realignment with and without LoRA adapters. Notably, without LoRA, the average downstream task transfer performance reduces by nearly 17 percentage points, with the most significant detrimental effect observed on NLI tasks with a ∼19 percentage point reduction.

The decision to utilize LoRA for decoder-only models in our experiments is driven by two distinct motivations. The primary constraint, as previously noted in our Limitations, is computational cost. Conducting full fine-tuning on a 9-billion parameter model over 16,000 realignment steps, followed by additional downstream fine-tuning, exceeded our available computational budget. Consequently, we followed Liu and Niehues (2025) in adopting adapter-based tuning.

A secondary motivation is the hypothesis that full-parameter realignment of a decoder-only model risks destabilizing its generative capabilities, potentially provoking language confusion. By employing adapter-based tuning, the core weights of the pre-trained backbone remain frozen, theoretically circumventing this instability. To empirically validate this assumption rather than relying on speculation, we conducted an ablation study comparing uniform realignment with and without LoRA using a smaller Gemma-2-2B model.

As illustrated in Table 5, the empirical results strongly support this hypothesis. The full fine-tuning approach (without LoRA) exhibits clear signs of catastrophic forgetting, resulting in significantly lower scores. Conversely, the low-rank LoRA updates successfully preserve the model's stability, maintaining robust performance across the 2 evaluated tasks.

C.5 Data and evaluation details

We follow the experimental setup of Nguyen et al. (2025), using OPUS-100 (Zhang et al., 2020a) and NLLB (Costa-Jussà et al., 2022) as realignment corpora and the same downstream benchmarks. We distinguish four language categories: HRLs, High-Resource Languages of Joshi (Joshi et al., 2020) class 5, MRLs, Medium-Resource Languages of Joshi class 3 and 4, LRLs seen, Low-Resource Languages that were seen during pre-training of the model, and LRLs unseen, Low-Resource Languages that were not seen during pre-training. The exact language composition of each group is provided in Appendix C.5. The distinction between seen and unseen LRLs is not made for Gemma, as we are unsure of what languages were included in its pre-training data. Table 6 reports, per Joshi resource class (Joshi et al., 2020), the number of languages evaluated, the range of parallel training sentences used for realignment, and the size range of each evaluation set. All 65 realignment languages are included. Rows where *Realignment n < Langs* (classes 0 and 1) contain these eval-only languages.

Nguyen et al. (2025) originally included AmericasNLI (Ebrahimi et al., 2022), which contains languages that are not available in the parallel dataset. For this very reason, we decided not to include it in our experiments.

All experiments are run with seeds of 31, 42 and 66.

C.6 The link between UCB and gradient-based approaches

This section provides a more detailed discussion about the conceptual similarities between the UCB-based and gradient-based approaches.

At each step of the outer loop, we've got a loss-per-language vector $L \in \mathbb{R}^{|\mathcal{L}_T|}$, where L_ℓ is the average realignment loss for language ℓ accumulated over the last k inner steps. The optimization problem that both methods are trying to solve is to find a language distribution p that maximizes the expected loss:

$$\max_p \sum_{\ell \in \mathcal{L}_T} p_\ell L_\ell \tag{5}$$

This is a linear optimization problem over the simplex, and its solution is to put all the mass on the language with the highest loss. However, this would lead to overfitting and neglecting other languages.

The gradient-based method addresses this indirectly by performing a gradient descent step instead of directly solving the optimization problem, which leads to a smoother update that doesn't put all the mass on a single language.

The UCB-based update, on the other hand, is expressed as follows:

$$p = \text{softmax}\left(\frac{L + c\sqrt{\frac{\ln t}{n+1}}}{\tau}\right) \tag{6}$$

If we ignore the exploration term, we get the closed-form solution of the following optimization problem:

$$\max_p \sum_{\ell \in \mathcal{L}_T} p_\ell L_\ell - \tau H(p) \tag{7}$$

Where $H(p)$ is the entropy of the distribution p. This is a regularized version of the original optimization problem, where the entropy term encourages exploration and prevents putting all the mass on a single language.

Therefore, both methods can be seen as trying to solve the same underlying optimization problem, but with different approaches to prevent overfitting and encourage exploration. The gradient-based method does this through a smooth update, while the UCB-based method does this through an explicit regularization term in the optimization objective.

This also explains why the UCB-based method converges faster in the early stages of training, as it has a more direct way to find the optimal distribution, while the gradient-based method might be slower to find the optimal distribution due to its indirect approach.

D Additional results

D.1 Dynamic Languages Weights Analysis

To better understand the dynamics of our proposed strategies, Figures 6 and 7 plot the evolution of language weights across training for XLM-R and Gemma 2 9B respectively. Both UCB and gradient-based methods oversample roughly the same top-5 set of languages per model, despite their different designs, and this ordering is consistent across seeds. For Gemma, the oversampled languages are all LRLs except Malay (ms). For XLM-R Large, they include mid-resource languages like Hebrew

		Realignment		NER		POS		NLI	
Joshi class	**Langs**	*n*	size range	*n*	size range	*n*	size range	*n*	size range
0 (LRL)*	3	3	1.9M–2.8M	5	966–1,613	5	599–646	6	750
1 (LRL)	16	16	181k–46.4M	15	100–2,235	8	146–642	14	600–5,010
2 (LRL)	10	10	107k–32M	9	100–1,883	8	47–713	7	600
3 (MRL)	17	17	227k–63.6M	17	1,000–10,000	13	425–4,127	5	2,984–5,010
4 (MRL)	13	13	534k–1M	13	1,000–10,000	13	449–8,973	4	5,010
5 (HRL)	6	6	1M	6	10,000	6	1,680–22,358	5	5,010
Total	**65**	**65**	107,296–63,581,148	**65**	—	**53**	—	**28**	—
Train (en)	1	-	-	1	20,029	1	12,570	1	392,702

Table 6: Per-Joshi-class language coverage and evaluation set size ranges, following Nguyen et al. (2025). *Langs* counts all evaluated languages in that class; *Realignment n* is the subset used for realignment (classes 0 and 1 include additional eval-only languages, hence *Realignment n* < *Langs*). LRL = low-resource (classes 0–2), MRL = medium-resource (3–4), HRL = high-resource (5). NER: WikiANN + MasakhaNER; POS: UDPOS + MasakhaPOS; NLI: XNLI, IndoNLI, Myanmar-XNLI, and AfriXNLI. *Ghomala is tentatively assigned to class 0; its Joshi class is absent from the taxonomy of Joshi et al. (2020).

EXP	NER (F1)	NLI (Acc)	POS (Acc)	AVG
XLM-R-Large				
Fine-tuning only	57.83 ±0.90	65.54 ±0.39	68.90 ±0.61	64.09 ±0.38
Uniform realignment	61.97 ±0.57 (+4.14)	67.70 ±0.34 (+2.16)	71.56 ±0.40 (+2.67)	67.08 ±0.13 (+2.99)
Most-URIEL	62.00 ±0.56 (+4.17)	67.98 ±0.32 (+2.44)	70.93 ±0.35 (+2.03)	66.97 ±0.37 (+2.88)
Gradient-based	**62.25 ±1.10 (+4.42)**	68.65 ±0.29 (+3.11)	**72.33 ±0.34 (+3.44)**	**67.75 ±0.31 (+3.66)**
UCB-based	61.93 ±0.79 (+4.09)	**68.94 ±0.02 (+3.40)**	72.33 ±0.42 (+3.44)	67.73 ±0.13 (+3.64)
Gemma-2-9b				
Fine-tuning only	36.09 ±1.18	63.30 ±3.67	49.09 ±0.94	49.49 ±1.42
Uniform realignment	36.51 ±1.35 (+0.42)	69.97 ±1.08 (+6.67)	48.22 ±4.70 (-0.87)	51.57 ±2.24 (+2.07)
Most-URIEL	**37.60 ±1.17 (+1.51)**	**70.11 ±1.01** (+6.81)	**51.83 ±0.85 (+2.74)**	**53.18 ±0.28 (+3.69)**
Gradient-based	34.54 ±3.94 (-1.55)	69.31 ±0.81 (+6.01)	50.15 ±1.41 (+1.06)	51.33 ±1.45 (+1.84)
UCB-based	37.06 ±0.30 (+0.97)	69.36 ±0.36 (+6.07)	50.06 ±5.97 (+0.97)	52.16 ±1.87 (+2.67)
mBERT				
Fine-tuning only	52.85 ±0.25	62.94 ±0.08	53.03 ±0.04	56.27 ±0.09
Uniform realignment	**56.33 ±0.23 (+3.48)**	68.82 ±0.06 (+5.88)	58.40 ±0.07 (+5.36)	**61.18 ±0.08 (+4.91)**
Most-URIEL	55.76 ±0.17 (+2.91)	67.92 ±0.35 (+4.99)	58.43 ±0.21 (+5.40)	60.70 ±0.15 (+4.43)
Gradient-based	55.39 ±0.65 (+2.54)	**69.05 ±0.10 (+6.11)**	58.72 ±0.37 (+5.69)	61.05 ±0.25 (+4.78)
UCB-based	56.05 ±0.47 (+3.20)	68.71 ±0.06 (+5.78)	**58.76 ±0.34 (+5.72)**	61.17 ±0.25 (+4.90)
mmBERT				
Fine-tuning only	45.32 ±1.97	58.64 ±1.99	60.21 ±0.44	54.72 ±1.30
Uniform realignment	51.16 ±1.20 (+5.83)	**68.99 ±0.30 (+10.34)**	63.48 ±1.18 (+3.27)	61.21 ±0.78 (+6.48)
Most-URIEL	51.81 ±0.43 (+6.48)	68.09 ±0.74 (+9.45)	63.14 ±0.25 (+2.94)	61.01 ±0.46 (+6.29)
Gradient-based	**53.59 ±2.56 (+8.27)**	68.64 ±0.16 (+10.00)	63.50 ±0.67 (+3.29)	**61.91 ±0.70 (+7.19)**
UCB-based	52.48 ±1.71 (+7.15)	68.26 ±0.82 (+9.62)	**63.69 ±0.29 (+3.48)**	61.48 ±0.76 (+6.75)
mDeBERTa-v3				
Fine-tuning only	66.88 ±0.22	**60.87 ±0.77**	71.35 ±0.26	66.36 ±0.30
Uniform realignment	68.33 ±0.05 (+1.45)	59.72 ±0.88 (-1.15)	**72.44 ±0.47 (+1.09)**	66.83 ±0.35 (+0.46)
Most-URIEL	68.09 ±0.16 (+1.21)	58.83 ±0.45 (-2.04)	71.34 ±0.56 (-0.00)	66.09 ±0.03 (-0.28)
Gradient-based	**68.79 ±0.16 (+1.91)**	60.04 ±1.26 (-0.83)	71.72 ±0.88 (+0.37)	66.85 ±0.25 (+0.49)
UCB-based	68.53 ±0.30 (+1.65)	60.30 ±0.87 (-0.57)	72.05 ±0.40 (+0.70)	**66.96 ±0.21 (+0.59)**
Llama 3.1 8b				
Fine-tuning only (LoRA)	40.36 ±2.72	56.55 ±2.63	29.98 ±0.62	42.30 ±0.33
Uniform realignment (LoRA)	41.50 ±6.10 (+1.14)	62.01 ±1.09 (+5.45)	31.79 ±2.01 (+1.81)	45.10 ±2.92 (+2.80)
Most-URIEL	42.58 ±2.53 (+2.22)	62.01 ±0.83 (+5.46)	31.86 ±1.92 (+1.88)	45.48 ±1.52 (+3.19)
Gradient-based (LoRA)	**43.24 ±3.10 (+2.88)**	61.50 ±1.76 (+4.94)	**32.67 ±0.72 (+2.69)**	**45.80 ±1.36 (+3.50)**
UCB-based (LoRA)	41.26 ±5.41 (+0.91)	**62.28 ±0.89 (+5.73)**	32.04 ±1.02 (+2.06)	45.20 ±2.41 (+2.90)

Table 7: Average performance across three random seeds for each task and across all three tasks, comparing three baselines with two dynamic sampling strategies. All realignment experiments use 16,000 realignment steps.

(he), low-resource ones like Tswana (tsn_Latn) and Tamil (ta), and typologically distant ones like Malay (ms) and Hindi (hi), though Hindi is Indo-European like English. Since this oversampling yields better downstream results, this supports our hypothesis that realignment benefits from oversampling LRLs.

The two methods differ in how the weights evolve. For XLM-R, UCB quickly converges to a stable distribution with slower subsequent dynamics (e.g., the gradual increase of Hindi), while the gradient-based method appears monotonically divergent without stabilizing, though some non-monotonic evolution is visible for Malay, especially with Gemma 2. This reflects a key limitation: gradient-based has no explicit exploration mechanism and may over-exploit early winners, whereas UCB explores by design.

More importantly, the gradient-based method is slower in early stages but suffers from strong momentum later. For XLM-R, Hebrew reaches a sampling probability above 0.04 in under 1k steps with UCB, but takes roughly 10× longer with gradient-based; it then keeps rising with gradient-based while slowly diminishing with UCB. This likely explains why UCB performs better on longer runs while gradient-based degrades at 32k steps (Figure 3). The slower early learning of the gradient-based method may stem from it being a single gradient step, whereas UCB's update closely resembles the closed-form solution to the regularized problem of maximizing the realignment loss (Appendix C.6).

D.2 Realignment Step Scaling

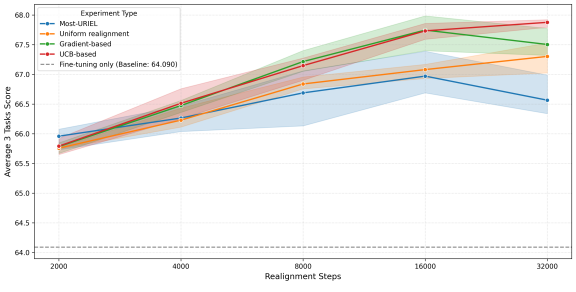


Figure 3: Scaling of average cross-lingual transfer performance for XLM-R

Most-URIEL achieves stronger gains than the other strategies at low compute budgets (2k steps), but the gap widens in favor of the dynamic strategies as steps increase (Figure 3). At 2,000 steps, Most-URIEL leads with an average score of roughly 66.0 — approximately 0.2 points ahead of the dynamic methods. This reflects the data-efficiency of typologically-informed static selection: with scarce compute, every step targets a pre-curated diverse language subset, consistent with the finding that linguistic diversity is a key driver of language subset quality for realignment (Nguyen et al., 2025). Dynamic methods, starting from a near-uniform distribution, seem to require a warm-up period for their weights to concentrate toward hard-to-align languages, incurring an early exploration cost. As training continues, however, the static selection's advantage erodes. At 16,000 steps, UCB and gradient-based sampling both reach an average of 67.73–67.75 (Table 1), surpassing Most-URIEL's 66.97 by nearly 0.8 points. This crossover directly reflects the rigidity of static distributions: since alignment difficulty shifts throughout training (see Figure 6), a distribution fixed before training begins becomes progressively misaligned with the model's evolving needs (Albalak et al., 2023). Most importantly, this shows that dynamic sampling scales better with training time, as it can adapt to the model's changing alignment needs, while static sampling cannot.

Gradient-based degrades at 32k steps, while UCB remains stable — pointing to UCB's greater robustness to training duration as a key advantage. Concretely, the gradient-based method drops from 67.75 at 16,000 steps to approximately 67.50 at 32,000, while UCB continues to improve to roughly 67.9. We attribute this to hyperparameter sensitivity: all hyperparameters were tuned via grid search at 16,000 steps, and gradient-based seems to have overfit on this setting. This brittleness echoes the behaviour observed on Gemma (Section 4), where gradient-based also underperformed — in both cases, performance degrades when conditions diverge from the tuning regime. UCB is less exposed to this issue: its update rule has a principled closed-form character (Appendix C.6) and its single exploration coefficient c proves less sensitive to the choice of training length.

Nevertheless, we find 16,000 realignment steps to provide the best trade-off between computational cost and performance, as doubling the training time to 32,000 steps yields only an additional gain of roughly 0.4 points. These findings provide practical guidance for selecting realignment strategies under different resource constraints.

D.3 Language-level win rate analysis

To complement the average performance analysis, we also examine the win rates of the different methods. This win rate is computed, given two methods A and B, as the percentage of runs, represented by a unique task, language, and seed, for which method A outperforms method B. This analysis allows us to understand how consistent the performance gains are across languages, and whether the improvements are driven by a few languages or are more widespread.

It is in the win-rate analysis that the positive impact of dynamic sampling becomes most apparent (Figure 4). Both dynamic sampling methods further improve cross-lingual transfer gains over the static baselines (uniform and Most-URIEL) while also enhancing language-wise performance. In particular, both dynamic sampling methods outperform uniform sampling by at least 61.9%. This is coherent with resource-level breakdown results, where we observed that dynamic sampling spreads the gains across more languages.

In the case of Gemma, all realignment methods win versus fine-tuning, and Most-URIEL is a surprisingly strong baseline, which mirrors the overall results. However, the win-rates show that UCB-based dynamic sampling still wins over uniform sampling, and is second only to Most-URIEL. Moreover, the gradient-based method falls behind. This suggests that a set of hyperparameters that works for one model will probably not be optimal for another one, since we tuned ours with XLM-R. As we already saw earlier, the UCB-based method generalizes better to other architectures than the gradient-based one.

For Gemma, realignment itself is highly effective, and while UCB is the most reliable dynamic approach, Most-URIEL sets a high bar that UCB doesn't clearly surpass. When working with a new model architecture, using a subset of languages might be a good starting point before trying to implement a dynamic sampling strategy.

We additionally present a win-rate comparison of our proposed UCB-based method across the four supplementary models (three encoder-only and one decoder-only). As illustrated in Figure 5, the UCB-based strategy establishes a formidable baseline, outperforming uniform realignment on the majority of models - with uniform sampling edging it out by a marginal 0.4% on mmBERT. The approach to most frequently surpass it (on 2 of the 4 models) is our alternative gradient-based method. Although the gradient-based strategy yields limited gains over the uniform sampling baseline on Gemma-2-9B, it achieves superior performance on both mmBERT and Llama 3.1 8B.

E Computational resources

All experiments with XLM-R Large and Gemma 2 9B were run on a single H100 80GB GPU. The total compute used for all experiments is estimated to be around 1,000 GPU hours.

A typical realignment run with XLM-R for 16k steps takes around 4.5 hours, while a typical run with Gemma 2 9B takes around 7 hours (not including fine-tuning).

F Licenses for artifacts used

Below is a list of the datasets under study:

- The AfriXNLI dataset (Adelani et al., 2025) has the Apache 2.0 license.
- The IndoNLI dataset (Mahendra et al., 2021) has the CC-BY-SA 4.0 license.
- The Myanmar-XNLI dataset (Htet and Dras, 2025) has the Apache 2.0 license.
- The MasakhaPOS dataset (Dione et al., 2023) has the MIT license.
- The MasakhaNER 2.0 dataset (Adelani et al., 2022) has the AFL 3.0 license.
- The OPUS-100 dataset (Zhang et al., 2020b) has no explicit license; it is a filtered subset of OPUS (Tiedemann, 2009), which aggregates translation corpora that is generally considered redistributable.
- The NLLB dataset (Team et al., 2022) has the ODC-By license.
- The XTREME-R benchmark suite (Ruder et al., 2021) does not have a unified license; it aggregates multiple datasets, each with its own license or terms of use, here are the ones we use:
  - The XNLI corpus (Conneau et al., 2018) has the CC BY-NC 4.0 license.
  - The UDPOS dataset (De Marneffe et al., 2021) has the CC0-1.0 license.

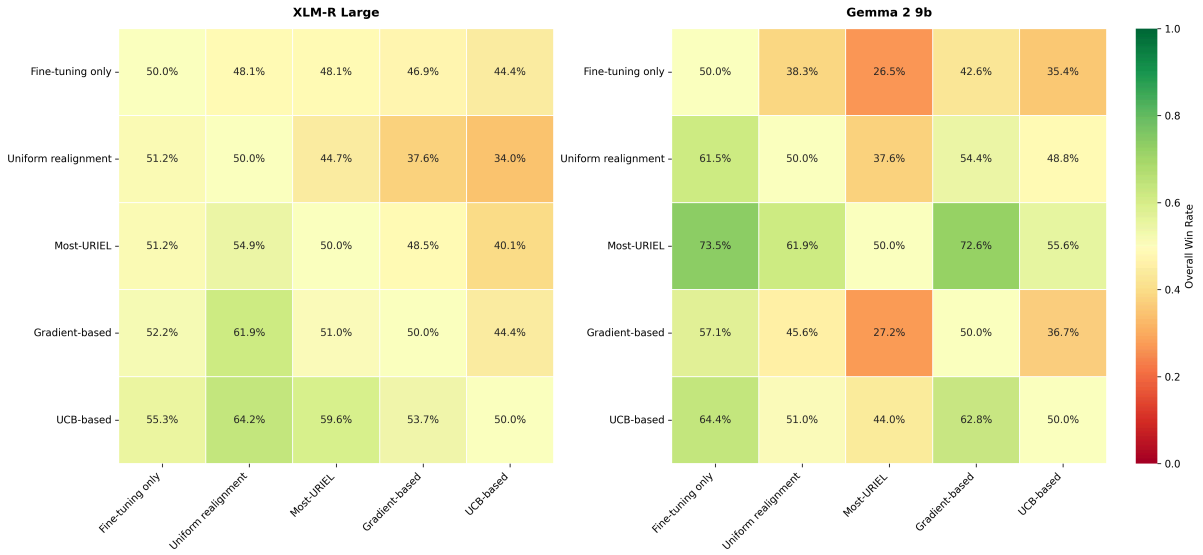


Figure 4: Win Rate, N=441 (all tasks, all languages, all seeds). The value of a cell indicates how many times the method in the row outperformed the method in the column.

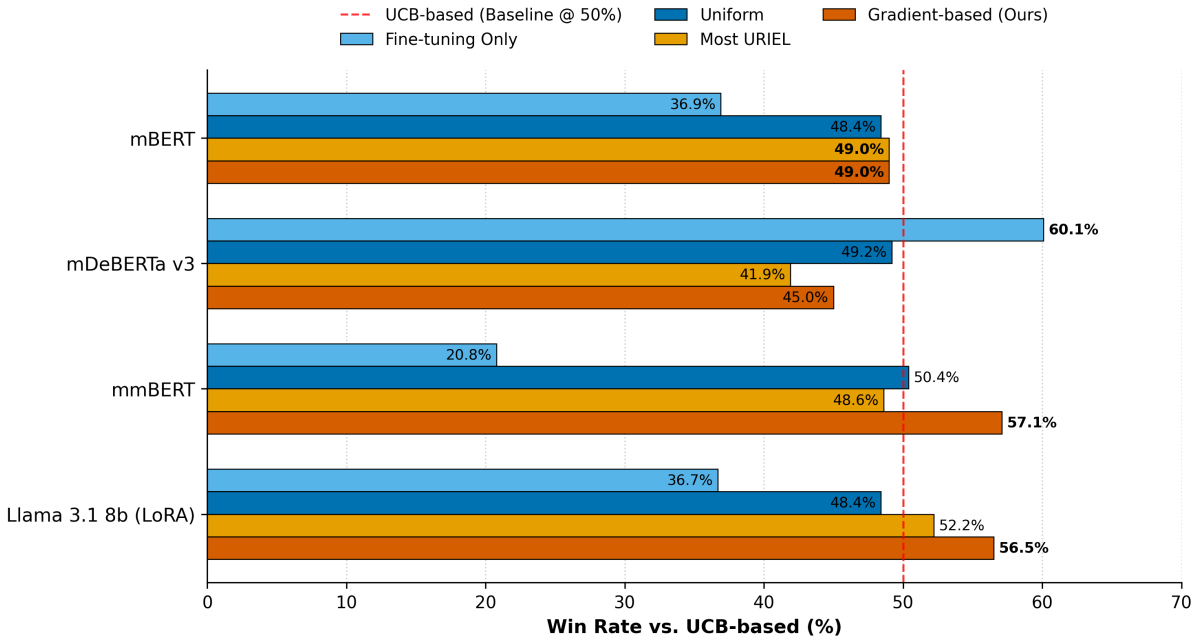


Figure 5: Win Rate comparing other experiments to our proposed UCB-based method for extra 3 encoder-only models (mBERT, mDeBERTa-v3 and mmBERT) and 1 decoder-only model (Llama 3.1 8b) on the same tasks, languages and seeds. Bold indicates the the best win-rate of each model. Uniform realignment cannot surpass UCB-based strategy in distributing realignment gain across more languages in most of the cases.

  - The WikiANN dataset (Pan et al., 2017) has the Apache 2.0 license.

Below is a list of the other artifacts under study:

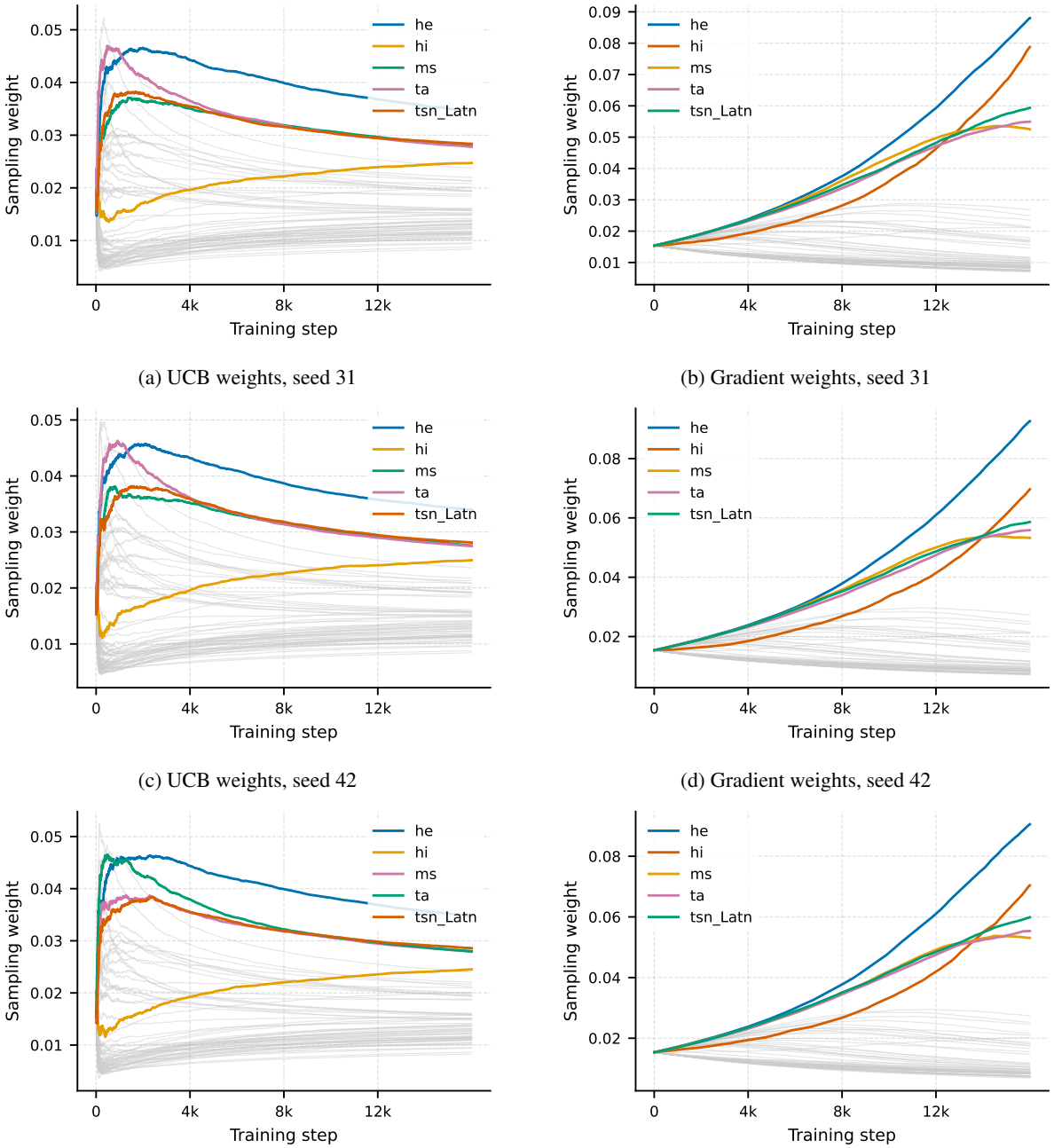
- The code for realignment comes from Nguyen et al. (2025) and has the MIT license.
- The weights of XLM-R Base (Conneau et al., 2020) have the MIT license.
- The weights of Gemma 2 9B (Team et al., 2024) have a dedicated Gemma License, which allows for research use and redistribution under certain conditions.

All artifacts were thus used in accordance with their open-source or non-commercial licenses.

G Use of AI

For the writing of this paper, AI was used for the following purposes: reformulate some text, autocomplete code, help with technical issues for Python code and LaTeX, and occasionally help with brainstorming and for generating code that would produce the figures.

All the ideas of the paper were proposed by the authors. All the text was originally written and revised by the authors, with some reformulation help from AI. All generated code was revised and edited by the authors. No figure was directly generated by AI.



(a) UCB weights, seed 31

(b) Gradient weights, seed 31

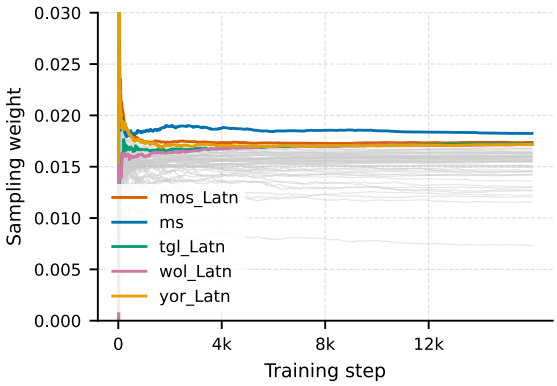
(c) UCB weights, seed 42

(d) Gradient weights, seed 42

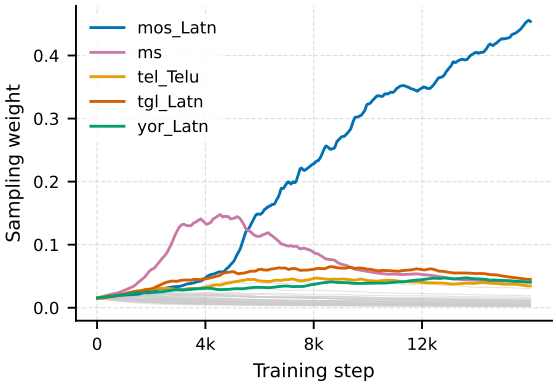
(e) UCB weights, seed 66

(f) Gradient weights, seed 66

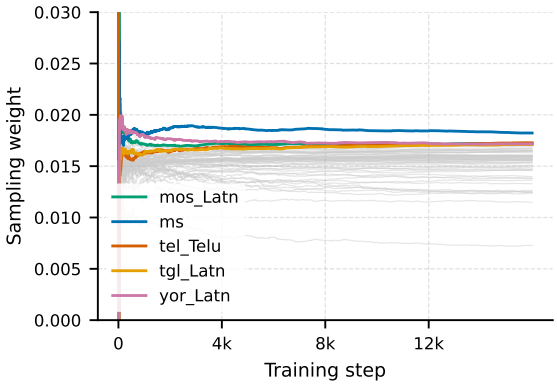
Figure 6: Learned weights for XLM-R Large across seeds for UCB (left) and gradient-based (right) selection strategies.



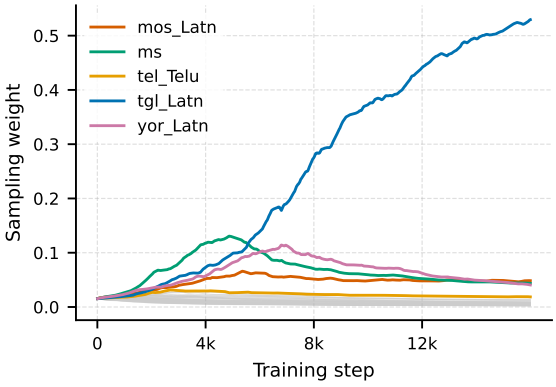
(a) UCB weights, seed 31



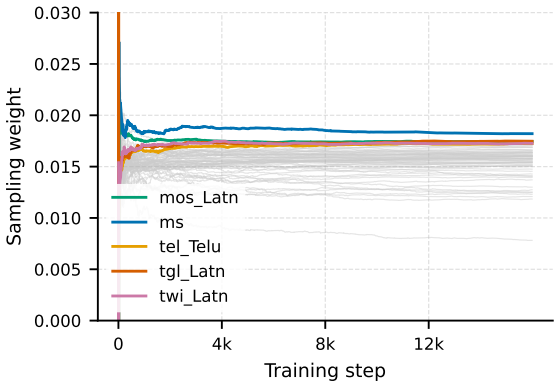
(b) Gradient weights, seed 31



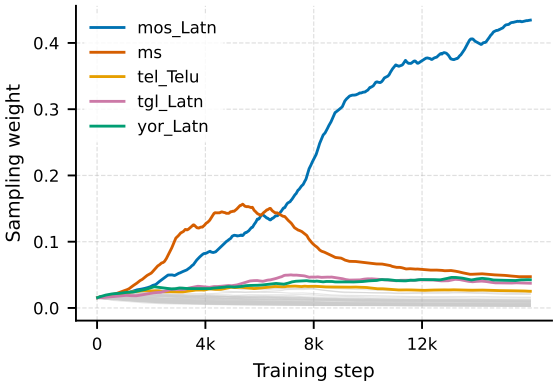
(c) UCB weights, seed 42



(d) Gradient weights, seed 42



(e) UCB weights, seed 66



(f) Gradient weights, seed 66

Figure 7: Learned weights for Gemma 2 9B across seeds for UCB (left) and gradient-based (right) selection strategies.