Controlling Language Confusion in Multilingual LLMs

Nahyun Lee[1,3] **Yeongseo Woo**[1] **Hyunwoo Ko**[2,3] **Guijin Son**[2,3]

Chungang University[1] OneLineAI[2] MODULABS[3]
naa012@cau.ac.kr spthsrbwls123@yonsei.ac.kr

Abstract

Large language models often suffer from language confusion, a phenomenon in which responses are partially or entirely generated in unintended languages. This critically degrades the user experience, especially in low-resource settings. We hypothesize that this issue stems from limitations in conventional fine-tuning objectives, such as supervised learning, which optimize the likelihood of correct tokens without explicitly penalizing undesired outputs such as cross-lingual mixing. Analysis of loss trajectories during pretraining further reveals that models fail to distinguish between monolingual and language-mixed texts, highlighting the absence of inherent pressure to avoid such confusion. In this work, we apply ORPO, which adds penalties for unwanted output styles to standard SFT, effectively suppressing language-confused generations. ORPO maintains strong language consistency, even under high decoding temperatures, while preserving general QA performance. Our findings suggest that incorporating appropriate penalty terms can effectively mitigate language confusion in multilingual models, particularly in low-resource scenarios.

1 Introduction

Scaling large language models has empirically delivered substantial gains in multilingual capabilities (Hurst et al., 2024; Cohere et al., 2025; Yang et al., 2025), across diverse tasks such as machine translation (Alves et al., 2024), summarization (Forde et al., 2024), and reasoning (Son et al., 2025). However, despite their growing capabilities, LLMs often suffer from language confusion (Marchisio et al., 2024), a failure mode in which outputs inadvertently blend multiple languages. This hampers real-world deployment of LLM systems as even the most minor language confusion may be critical to user experience (Son et al., 2024a). This issue is particularly pronounced in low-resource settings, where limited supervision exacerbates cross-lingual interference (Arivazhagan et al., 2019; Wang et al., 2023).

However, little research has been conducted on *why* such behavior may happen. In this work, we draw inspiration from the training methodology proposed by Hong et al. (2024), which applies supervised fine-tuning to preferred generation styles while imposing penalties on disfavored ones.

In this work, we conduct two experiments to investigate whether language confusion arises from the absence of an explicit penalty against undesired languages.

First, we track the training loss of two model families (SmolLM2 (Allal et al., 2025) and OLMo2 (OLMo et al., 2024)) throughout their pretraining process. In both cases, the loss of language-confused outputs steadily decreases over time, indicating that the models do not learn to disfavor confused generations. Additionally, by using ORPO (Hong et al., 2024) for an additional three epochs of fine-tuning, we show that introducing an explicit penalty against unwanted languages effectively restricts language confusion.

2 Preliminaries

2.1 Related Works

What is language confusion? Language confusion, also known as language mixing or code-mixing, occurs when two or more languages are mixed within a single utterance (Chen et al., 2024; Yoo et al., 2024). This phenomenon is particularly prevalent in low-resource languages (Arivazhagan et al., 2019) and even appears in state-of-the-art models (u/VictorRM, 2025). Diverse discussions have emerged regarding language confusion. Although it can sometimes support multilingual transfer (Wang et al., 2025), mixed-language responses may undermine user experience, as they can be perceived as signs of incompetence (Son et al., 2024a).

2.2 Quantifying Language Confusion

Measurement of language confusion can be challenging, as LLM judges (Zheng et al., 2023) remain unreliable (Son et al., 2024b), and rule-based methods cannot distinguish genuine confusion from legitimate uses of foreign language (e.g., abbreviations). In this work, we leverage two metrics Word Precision Rate (WPR) and Language Precision Rate (LPR) proposed by Marchisio et al. (2024).

WPR computes the overall fraction of tokens produced in the target language, offering a granular view of how consistently a model sticks to one language. Where $\mathcal{T} = \bigcup_{i=1}^{N} T_i$ is the set of all valid tokens across N outputs, WPR is defined as:

$$\frac{\left|\{\, t \in \mathcal{T} : \text{is_Korean}(t) \}\right|}{|\mathcal{T}|} \tag{1}$$

LPR counts the proportion of sentences in which at least 90% of tokens belong to the target language, thereby penalizing any cross-lingual intrusions. Where $I(\cdot)$ denotes the indicator function and s_i the i-th sentence, LPR is defined as:

$$\frac{1}{N} \sum_{i=1}^{N} I\Big(\frac{\left|\{\, t \in s_i : \text{is_Korean}(t) \}\right|}{\left|\{\, t \in s_i : \text{is_valid}(t) \}\right|} \;\geq\; 0.9 \Big) \tag{2}$$

Additionally, as noted above, rule-based metrics alone cannot distinguish true language confusion from minor lexical variations, such as numerals, named entities, or common loanwords. Therefore, alongside WPR and LPR, we also report the proportion of responses with WPR and LPR exceeding 0.9. Empirically, we observe that many such responses remain perfectly acceptable sentences containing a few legitimate English terms. For examples of sentences with varying WPR and LPR levels, see Appendix D.

3 Experimental Setup

3.1 Dataset Preparation

To facilitate pairwise preference learning, we constructed instruction-centered triplet datasets. Each triplet comprises a Korean prompt *(input)*, a fully Korean response *(chosen)*, and an alternative response exhibiting code-mixing or a full unexpected language *(rejected)*.

We constructed three multilingual datasets based on existing Korean corpora, each designed to represent a different form of language confusion. The

```
### Input: 여행 준비를 위한 최고의 팁은 무엇입니까?

### Chosen: 1.간식과 물과 같은 물품이 충분한지 확인하십시오. 2.경로를
미리 계획하여 목적지와 도착하는 데 걸리는 시간을 알 수 있습니다. 3.짐은
가볍게 하되 재킷, 모자, 장갑 등을 준비하십시오.

### Rejected: 1.Make sure you have enough supplies, such as
snacks and water. 2.Plan your route in advance so that you
know where you're going and how long it will take to get there.
3.Pack light but still be prepared with items like
jackets, hats, gloves, etc.
```

Figure 1: Dataset structure (OIG, Chosen-Rejected pair)

OIG dataset (LAION, 2022; Heegyu, 2023) and HC3 dataset (Guo et al., 2023; Na, 2023) pair Korean prompts with rejected responses written entirely in English. In contrast, the KoAlpaca dataset (Beomi, 2023) introduces more nuanced confusion by synthetically injecting translated English or Chinese tokens into Korean outputs, resulting in code-mixed responses. Additional preprocessing and filtering steps are described in Appendix A.

3.2 Experiment Setup

We fine-tuned two publicly available instruction-tuned language models: SmolLM2-1.7B (Allal et al., 2025) and OLMo2-7B (OLMo et al., 2024), selected for their ability to generate Korean text among lightweight open source models. Detailed training configurations are provided in Appendix B.

3.3 Evaluation Protocol

We evaluate three model variants: **Base**, the original instruction-tuned model; **SFT**, supervised fine-tuned on Korean prompt–response pairs from the OIG dataset; and **ORPO**, fine-tuned using Odds Ratio Preference Optimization, on the same dataset.

4 Main Results

Prior work shows LLMs default to high-frequency, dominant-language tokens when uncertain, causing language confusion (Marchisio et al., 2024). We hypothesize that the standard next-token prediction objective exacerbates this bias: softmax focuses probability mass on the correct token but does not explicitly penalize cross-lingual mixing.

4.1 Loss-Based Diagnostic: Do LLMs Penalize Language Mixing?

We begin with the observation that, during pretraining, neither SmolLM2 (Allal et al., 2025) model learns to penalize language confusion, as shown by their loss trajectories in Figure 2.

	Model	SmolLM2-1.7B						OLMo2-7B					
	Temperature	0.7		1.0		1.2		0.7		1.0		1.2	
		Base	ORPO	Base	ORPO	Base	ORPO	Base	ORPO	Base	ORPO	Base	ORPO
Metric	WPR > 0.9 ratio	96.1%	100.0%	94.3%	100.0%	81.4%	100.0%	96.3%	99.8%	91.8%	99.9%	7.5%	99.0%
	LPR > 0.9 ratio	92.6%	99.9%	88.5%	100.0%	71.2%	99.9%	71.2%	99.7%	46.0%	99.8%	0.5%	96.8%
	Average WPR	0.9821	0.9999	0.9696	1.0	0.8953	0.9999	0.9818	0.9998	0.9576	0.9998	0.6799	0.9962
	Average LPR	0.9681	0.9996	0.9496	1.0	0.8434	0.9999	0.9379	0.9992	0.8684	0.9995	0.3044	0.9881

Table 1: Comparison of SmolLM2 and OLMo2 models across temperatures (Base vs. ORPO). All metrics are higher is better: higher values indicate stronger language consistency.

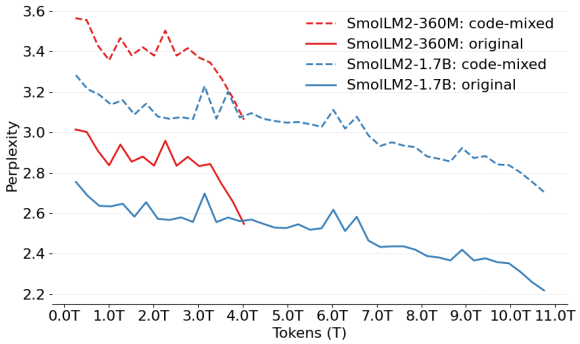

Figure 2: Average loss for monolingual and code-mixed responses across training tokens (SmolLM2)

In principle, a model that internalizes a robust linguistic preference should learn to assign lower loss to coherent Korean-only generations while preserving relatively higher loss for language-confused outputs. Contrary to expectations, we observe a monotonic decrease in loss for both chosen and rejected responses. This trend may suggest that, in the absence of explicit preference signals, models eventually learn to prefer *any* sequence of tokens they have seen during training, without distinguishing linguistically coherent and code-mixed outputs. Such behavior persists up to the 7B scale, suggesting that model size alone cannot resolve the issue. See Appendix C for results of OLMo2 models.

4.2 Generation-level evaluation: WPR and LPR Comparison

To evaluate the effectiveness of preference-based tuning method, we compare the generation performance of the Base and ORPO-tuned models using WPR and LPR under varying decoding temperatures. Each model generated responses for the same set of 1,000 prompts, repeated three times per prompt, and all reported scores are averaged across the three generations.

As summarized in Table 1, we observe the following trends:

- **ORPO-tuned models consistently outperform the Base models**, achieving near-perfect WPR and LPR even at high temperature settings (up to 1.2).
- **Temperature significantly impacts the Base models**. For instance, average LPR of the OLMo2 base model plummets to 0.3044 at a temperature of 1.2, indicating a severe degradation of linguistic consistency without preference-based fine-tuning.

5 Additional Results

5.1 Comparison with other fine-tuning methods

To evaluate how ORPO compares to other standard fine-tuning approaches, we conducted additional experiments using Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) under identical conditions.

Detailed results for both SmolLM2 and OLMo2 are presented in Appendix E. Across both model families, ORPO consistently achieves high WPR and LPR scores, matching or slightly exceeding SFT and substantially outperforming DPO.

5.2 Do fine-tuned models internalize penalties?

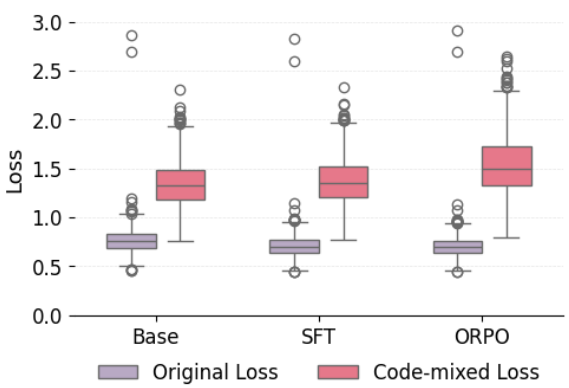

Figure 3: Loss of SmolLM2 models across tuning methods for both original and code-mixed responses

To further investigate whether preference-based learning offers additional internal modeling advantages, we conduct a loss-based diagnostic analysis on the evaluation subset HC3 and compare the loss between original *(chosen)* and code-mixed *(rejected)* responses.

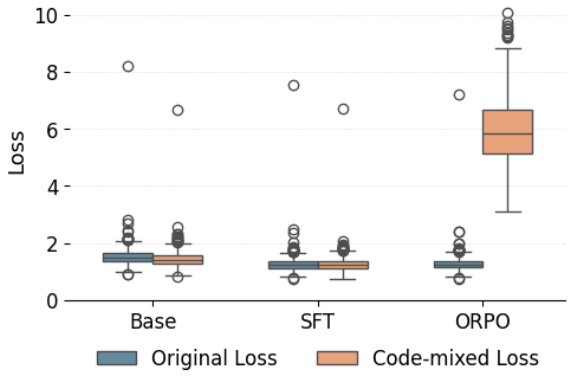

Figure 4: Loss of OLMo2 models across tuning methods for both original and code-mixed responses

We found that ORPO assigns significantly higher loss to code-mixed responses compared to other models, indicating stronger penalization of language-confused outputs. On the HC3 evaluation set, ORPO yields an average delta loss of 0.8379 for SmolLM2 and 4.6778 for OLMo2-both the highest among all fine-tuning methods. This increased separation suggests that ORPO fine-tuning more effectively reinforces internal preferences for linguistically consistent outputs, enabling more reliable discrimination between coherent and code-mixed generations (Figure 3 and 4).

5.3 Does ORPO Fine-Tuning Lead to a Trade-off in General QA Capabilities?

We assess whether ORPO fine-tuning, which mitigates language confusion, adversely affects general performance by evaluating our models on the HAE-RAE benchmark—a Korean multiple-choice QA suite covering general knowledge, history, loanwords, and rare vocabulary (Son et al., 2023). We omit more challenging reasoning benchmarks due to the modest size of our models and limited training data. We compared three model variants: Base, SFT and ORPO fine-tuned model.

Figure 5 reports the average accuracies in all subcategories for the SmolLM2 and OLMo2 models. The results show no significant performance degradation in the three tuning methods.

These findings suggest that neither SFT nor ORPO introduces measurable harm to general QA capabilities. In particular, ORPO maintains general QA performance while reducing language confusion.

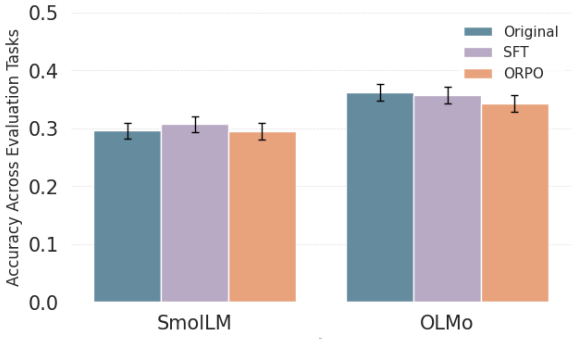

Figure 5: Average accuracy across training methods for SmolLM2 and OLMo2.

6 Conclusion

This work investigates the underlying causes of language confusion in multilingual large language models and empirically demonstrates that penalizing undesired languages via preference optimization is an effective method for suppressing such behavior.

Our primary contribution is the demonstration that preference-based fine-tuning offers a highly effective solution. By fine-tuning models to prefer monolingual responses over language-confused ones, we achieve robust linguistic consistency without compromising general question-answering capabilities.

These results suggest that incorporating explicit preference signals during fine-tuning provides a promising approach for reinforcing linguistic fidelity in multilingual settings. Moreover, we suggest that future research may explore the use of penalty terms even in the pretraining phase to penalize language confusion earlier in the training effectively.

Limitations

While our findings demonstrate the effectiveness of ORPO for mitigating language confusion, we acknowledge several limitations in this study.

First, our analysis does not include a sensitivity analysis of ORPO's hyperparameters. We used a fixed value ($\beta = 0.1$) based on the original ORPO paper. Future work should explore how varying this hyperparameter affects the trade-off between linguistic fidelity and general task performance.

Second, our experiments were conducted primarily on Korean-centric datasets and two specific model families (SmolLM2 and OLMo2). Although the results are strong, further research is needed to ascertain whether our findings generalize to other languages and other model architectures.

Third, we did not perform an in-depth analysis of why ORPO consistently outperforms DPO. Further investigation is needed to fully understand the optimization dynamics behind this difference.

Finally, although we have detailed our experimental setup and dataset construction, we have not yet released the code and training artifacts. To facilitate reproducibility, we plan to make all code and training materials publicly available upon publication.

Acknowledgements

This research was supported by Brian Impact Foundation, a non-profit organization dedicated to the advancement of science and technology for all.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, and 1 others. 2025. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, and 1 others. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Beomi. 2023. Koalpaca: Korean instruction-tuning dataset.

Yiyi Chen, Qiongxiu Li, Russa Biswas, and Johannes Bjerva. 2024. Large language models are easily confused: A quantitative metric, security implications and typological analysis. *arXiv preprint arXiv:2410.13237*.

Team Cohere, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, and 1 others. 2025. Command a: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*.

Jessica Zosa Forde, Ruochen Zhang, Lintang Sutawika, Alham Fikri Aji, Samuel Cahyawijaya, Genta Indra Winata, Minghao Wu, Carsten Eickhoff, Stella Biderman, and Ellie Pavlick. 2024. Re-evaluating evaluation for multilingual summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19476–19493, Miami, Florida, USA. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Heegyu. 2023. Oig-small-chip2-ko. https://huggingface.co/datasets/heegyu/OIG-small-chip2-ko.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

LAION. 2022. Open instruction generalist (oig) dataset. https://laion.ai/blog/oig-dataset/.

Kelly Marchisio, Wei-Yin Ko, Alexandre Bérard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in llms. *arXiv preprint arXiv:2406.20052*.

Yohan Na. 2023. Hc3-ko: Korean human chatgpt comparison corpus. https://huggingface.co/datasets/nayohan/HC3-ko. Accessed: 2025-05-17.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, and 1 others. 2024. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. Linguistic generalizability of test-time scaling in mathematical reasoning. *arXiv preprint arXiv:2502.17407*.

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024a. Llm-as-a-judge & reward model: What they can and cannot do. *arXiv preprint arXiv:2409.11239*.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jaecheol Lee, Je Won Yeom, Jihyu Jung, Jung Woo Kim, and Songseong Kim. 2023. Hae-rae bench: Evaluation of korean knowledge in language models. *arXiv preprint arXiv:2309.02706*.

Guijin Son, Dongkeun Yoon, Juyoung Suk, Javier Aula-Blasco, Mano Aslan, Vu Trong Kim, Shayekh Bin Islam, Jaume Prats-Cristià, Lucía Tormo-Bañuelos, and Seungone Kim. 2024b. Mm-eval: A multilingual meta-evaluation benchmark for llm-as-a-judge and reward models. *arXiv preprint arXiv:2410.17578*.

u/VictorRM. 2025. O3 thinks in chinese for no reason randomly. Reddit, r/OpenAI. Accessed 2025-05-19.

Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, AiTi Aw, and Nancy F Chen. 2023. Seaeval for multilingual foundation models: From cross-lingual alignment to cultural reasoning. *arXiv preprint arXiv:2309.04766*.

Zhijun Wang, Jiahuan Li, Hao Zhou, Rongxiang Weng, Jingang Wang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. 2025. Investigating and scaling up code-switching for multilingual language model pre-training. *arXiv preprint arXiv:2504.01801*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Haneul Yoo, Cheonbok Park, Sangdoo Yun, Alice Oh, and Hwaran Lee. 2024. Code-switching curriculum learning for multilingual transfer in llms. *arXiv preprint arXiv:2411.02460*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Dataset preprocessing

KoAlpaca (Code-Mixed Rejection): We constructed this dataset using the KoAlpaca[1] corpus, a Korean instruction-tuning dataset modeled after Stanford Alpaca (Beomi, 2023). Each triplet contains a Korean instruction, a fully Korean chosen response, and a synthetically generated code-mixed rejected response, created by injecting randomly selected English or Chinese tokens—translated via the Google Translate API—at random word-level positions.

To ensure high linguistic purity, we applied the following preprocessing steps: (1) filtered for chosen responses written entirely in Korean, guaranteeing a WPR and LPR of 1.0; (2) applied string normalization (e.g., whitespace trimming) to instruction, chosen, and rejected fields.

OIG (Fully English Rejection): We constructed a triplet dataset using the OIG-small-chip2-ko[2] corpus, which contains over 210K instruction-response pairs translated into Korean from the original English OIG dataset (LAION, 2022). Each triplet comprises a Korean instruction, a fully Korean chosen response, and a fully English rejected response. This dataset is designed to evaluate the model's ability to distinguish between clearly separated linguistic domains.

We applied several preprocessing steps to improve data quality: (1) applied string normalization; (2) filtered for chosen responses containing only Korean text; (3) discarded samples where the length ratio between chosen and rejected responses fell outside the range of 0.4 to 2.0; (4) removed duplicate instructions. Each dataset contains approximately 10,000 instruction-response triplets, selected for linguistic consistency and diversity.

HC3 (Fully English Rejection): We also constructed dataset using the HC3-ko[3], which contains 24.3k instruction pairs, each containing a human-written and a GPT-generated response, translated into Korean (Guo et al., 2023; Na, 2023).

Each triplet contains a Korean instruction, a fully Korean chosen response, and a synthetically generated code-mixed rejected response. This dataset is designed to evaluate the model's generalizing ability to use the unseen data during training.

We applied several preprocessing steps to improve data quality: (1) applied string normalization; (2) filtered for chosen responses containing only Korean text; (3) discarded samples where the length ratio between chosen and rejected responses fell outside the range of 0.4 to 2.0; (4) removed duplicate instructions. (5) removed responses exhibiting generation failures caused by the language model, such as repeated phrases or malformed outputs due to server errors.

[1] https://huggingface.co/datasets/beomi/KoAlpaca-v1.1a

[2] https://huggingface.co/datasets/heegyu/OIG-small-chip2-ko

[3] https://huggingface.co/datasets/nayohan/HC3-ko

B ORPO Training Configuration

Table 2 outlines the training configuration used for ORPO fine-tuning. Both SmolLM2-1.7B and OLMo-2-1124-7B were trained for 3 epochs with a global batch size of 128. ORPO's weighting coefficient β was set to 0.1 across experiments, and training was performed using the DeepSpeed ZeRO-2 framework.

Parameter	SmolLM2-1.7B (ORPO)	OLMo2-7B (ORPO)
GPUs	A6000 × 1	H100 × 2
Max sequence length	8192	4096
Micro batch size	8	8
Gradient accumulation	16	8
Global batch size	128	128
Training steps	223	223
Epochs	3	3
ORPO β value	0.1	0.1
Optimizer	AdamW	AdamW
Framework	DeepSpeed ZeRO-2	DeepSpeed ZeRO-2

Table 2: Training configuration for ORPO fine-tuning on SmolLM2 and OLMo2 models.

C Average loss tracking for OLMo2

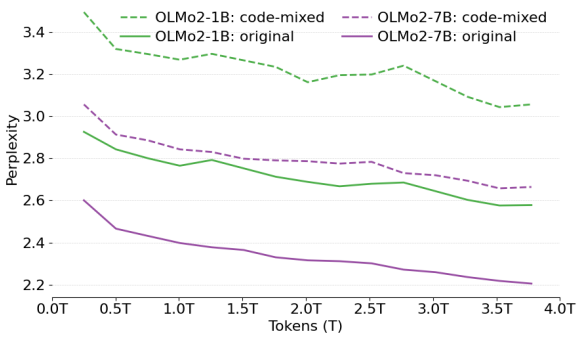

Figure 6: The average loss of original (monolingual) and code-mixed responses across training checkpoints for OLMo2 models.

To assess whether the failure to penalize language confusion generalizes across architectures, we also tracked the loss trajectories of OLMo2 models (1B and 7B) throughout pretraining. As shown in Figure 6, both original and code-mixed responses exhibit a steady decrease in loss, mirroring the trend observed in SmolLM2 (Figure 2). Despite the increase in model capacity, the gap between two responses does not widen. This suggests that pretraining objectives alone may not induce meaningful linguistic preferences.

D Samples of different levels of WPR and LPR

To enable interpretable comparisons across models, we report the proportion of generations that exceed a threshold of 0.9 for both WPR and LPR. This threshold was chosen based on manual inspection by a native Korean speaker, who reviewed a large number of generated samples and heuristically identified 0.9 as a practical cutoff that separates mostly monolingual responses from visibly code-mixed ones. This level of tolerance allows minor lexical variation (e.g., loanwords, numerals) while still maintaining strong target-language alignment. It also aligns with real world expectations for language consistency, particularly in Korean, where partial foreign-language inclusions are not uncommon but still undesirable in many contexts. Representative examples illustrating this thresholding effect are shown in Figure 7.

E Generation-level evaluation: other models

In addition to ORPO, we evaluate two other fine-tuning methods: Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) across multiple decoding temperatures and model families (SmolLM2, OLMo2).

Direct Preference Optimization (DPO) is a preference-based tuning method that trains models to maximize the log-probability margin between preferred and rejected responses (Rafailov et al., 2023).

Table 3 describes the detailed training configurations used for DPO fine-tuning. All settings were selected to closely match the original DPO implementation where possible.

Table 4 and Table 5 summarize the generation performance of each model across three decoding temperatures (0.7, 1.0, 1.2) and three fine-tuning methods (SFT, DPO, ORPO). We report four key metrics: the ratio of outputs with WPR > 0.9, LPR > 0.9, average WPR, and average LPR.

Across both model families, ORPO consistently outperforms DPO and performs on par with or slightly better than SFT in terms of language fidelity. In particular, ORPO maintains near-perfect WPR and LPR values across all temperature settings, while DPO exhibits significant degradation at higher temperatures, most notably on the OLMo2 model at temperature 1.2 (LPR > 0.9 ratio drops to

52.1%. SFT remains relatively stable across temperatures.

Parameter	SmolLM2-1.7B (DPO)	OLMo2-7B (DPO)
GPUs	A6000 × 1	A6000 × 4
Dataset size	10,000	10,000
Max sequence length	8192	4096
Micro batch size	8	4
Gradient accumulation	8	4
Global batch size	64	64
Training steps	467	467
DPO β value	0.1	0.1
Optimizer	RMSprop	RMSprop
Framework	DeepSpeed ZeRO-2	DeepSpeed ZeRO-2

Table 3: Training configuration for DPO fine-tuning on SmolLM2 and OLMo2 models.

Table 4: Performance of SmolLM2 across temperature and tuning methods (SFT, DPO, ORPO)

Metric	temperature = 0.7			temperature = 1.0			temperature = 1.2		
	SFT	**DPO**	**ORPO**	**SFT**	**DPO**	**ORPO**	**SFT**	**DPO**	**ORPO**
WPR > 0.9 ratio	99.9%	94.2%	100.0%	100.0%	96.9%	100.0%	100.0%	95.0%	100.0%
LPR > 0.9 ratio	99.8%	92.3%	99.9%	100.0%	94.4%	100.0%	99.7%	90.5%	99.9%
Average WPR	0.9998	0.9760	0.9999	1.0000	0.9857	1.0000	0.9998	0.9823	0.9999
Average LPR	0.9994	0.9705	0.9996	1.0000	0.9780	1.0000	0.9993	0.9629	0.9999

Table 5: Performance of OLMo2 across temperature and tuning methods (SFT, DPO, ORPO)

Metric	temperature = 0.7			temperature = 1.0			temperature = 1.2		
	SFT	**DPO**	**ORPO**	**SFT**	**DPO**	**ORPO**	**SFT**	**DPO**	**ORPO**
WPR > 0.9 ratio	99.8%	99.5%	99.8%	99.9%	99.4%	99.9%	99.1%	94.4%	99.0%
LPR > 0.9 ratio	99.7%	92.7%	99.7%	99.8%	89.4%	99.8%	96.8%	52.1%	96.8%
Average WPR	0.9996	0.9959	0.9998	0.9998	0.9938	0.9998	0.9970	0.9649	0.9962
Average LPR	0.9988	0.9847	0.9992	0.9997	0.9791	0.9995	0.9915	0.8897	0.9881

Input	Generated	WPR	LPR
임신 중 언제부터 수영을 할 수 있나요? 안녕하세요. 제 가족 중에 임신하신 분이 계시는데 수영을 하고자 하셔서요. 몇 개월부터 수영이 가능한지 궁금합니다.	임신 1달부터 수영이 가능합니다. 하지만, 수영 중 피부와 기관계의 건강을 위해, 수영 시간이 짧게 시작하고, 수영 및 노출 시간을 조절하여 피부에 부담을 최소화하는 것이 좋습니다. 또한, 임신 중에 수영을 하는 것은 개인의 건강 상태, 임신의 진화 상태, 및 관련 건강 문제 등을 고려해야 합니다.	1.0	1.0
개량 발명과 회피 발명은 무엇인가요? 또한 이러한 발명은 누구에 의해 할 수 있을까요? 마지막으로 원천 기술은 무엇인지 설명해주세요.	개량 발명(改良發明)은 기존 발명의 기능이나 성능을 향상시키기 위해 수정·보완한 기술을 말합니다. 회피 발명(回避發明)은 기존 특허를 침해하지 않도록 우회적인 방법으로 개발한 기술이며, 연구개발자(R&D)나 발명가에 의해 이루어질 수 있습니다. 반면, 원천 기술(源泉技術)은 특정 기술 분야에서 새로운 paradigm을 제시하는 근본적인 기반 기술로, 주로 연구소, 대학, 또는 대기업의 연구진에 의해 개발됩니다.	0.9	0.67
건전지의 플러스 극과 마이너스 극의 역할은 무엇인가요?	건전지의 플러스 극(positive terminal)은 전자를 받는 역할을 하며, 이는 전기 회로에서 전류가 흐르는 방향의 starting point입니다. 반대로, 마이너스 극(negative terminal)은 전자를 내보내는 역할을 하며, 에너지가 회로를 통해 흐르는 것을 가능하게 합니다. 이러한 양극(anode)과 음극(cathode)의 역할은 전기적 potential 차이를 기반으로 작동합니다. 건전지를 사용할 때는 polarity를 정확히 확인하는 것이 중요합니다.	0.8	0.0
커피와 설탕과 프림을 섞었을 때 흔들면 커피가 위로 올라오는 이유는 무엇인가요?	가장 큰 이유는 설탕, 프림, 커피의 밀도 차이 때문입니다. 설탕과 프림은 커피보다 밀도가 높아 아래로 가라앉습니다. 그 과정에서 커피는 상대적으로 위로 밀려 올라가게 됩니다. This phenomenon is caused by the difference in density among the components. As heavier particles sink, lighter coffee is displaced upward through convection-like motion.	0.5	0.6

Figure 7: Samples of generated responses at varying WPR and LPR levels