PROM: Pivoted and Regulated Optimization for Multilingual Instruction Learning

Jaeseong Lee[1], **Seung-won Hwang**[1]*, **Hojin Lee**[2], **Yunju Bak**[2], **Changmin Lee**[2]
[1]Computer Science and Engineering, Seoul National University
[2]Kakao Corp.
{tbvj5914,seungwonh}@snu.ac.kr
{lambda.xprime,juliet.bak,louie.m}@kakaocorp.com

Abstract

Large language models (LLMs) have become standard for natural language generation tasks, with instruction-tuning enhancing their capabilities. However, the lack of instruction-tuning datasets in languages other than English limits their application to diverse languages. To address this, researchers have adapted English-centric LLMs to other languages by appending English tuning data with its translated pair. However, we observe negative interference between the two. To resolve this, our contribution is identifying English as an internal pivot language, which disentangles the use of English and target language data. Moreover, to better generalize for under-represented languages, we regulate the proposed objective. Experiments across 9 different languages demonstrate the effectiveness of our approach on multiple benchmarks. The code is publicly available for further exploration.[1]

1 Introduction

Recently, large language models (LLMs) became a de-facto standard for various natural language generation tasks (OpenAI, 2023; Touvron et al., 2023; Jiang et al., 2024). Moreover, careful instruction-tuning (Wang et al., 2023) improves the LLMs to be more powerful.

However, due to the lack of instruction tuning datasets in other languages, most of instruction-tuned LLMs remain English-centric, hindering the application to 6500+ existing languages (Austin and Sallabank, 2011). Existing solutions thus propose to adapt English-centric LLMs into a monolingual target language model: Instructions in the target language are either unseen, or under-represented in pretraining, for which the existing solution translates a high-quality English instruction tuning, to pair with its translation in the target language (Zhu et al., 2023; Ranaldi et al., 2023).

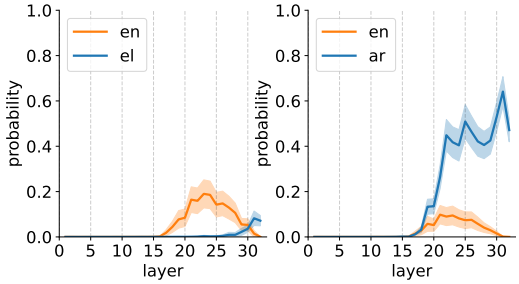

Figure 1: Language on the left shows pivoted behavior, as argued in Wendler et al. (2024). However, we find that argument does not hold in some languages (right).

Despite the expected performance gain from expanding the training set, our first contribution is observing otherwise, that negative interference (Conneau et al., 2020; Wang et al., 2020) exists between the original and translated pair. (Section 3.2).

To overcome this, we devise a *pivoted* objective that disentangles English and target language data in training, to alleviate such interference. Specifically, we are inspired by a recent finding that English-centric LLMs generate in English first and then convert the output into the target language (Wendler et al., 2024; Zhao et al., 2024). This implies that we can design two separate objectives, the first objective using English data for generating the representation corresponding to the English version of the next token at the middle of the layers, then another objective using target data for gradually converting into the representation for the target language.

While such disentangled objectives are effective in many languages, we find they fail to generalize well to under-represented languages, where we observe the pivoted behavior reported by Wendler et al. (2024) may not hold. To illustrate, Figure 1 contrasts language where pivoted assumption holds (left) and not (right), selected for illustration from our empirical studies reported in Appendix: Following (Wendler et al., 2024), the x-axis in the

* Corresponding author
[1]https://github.com/thnkinbtfly/PROM

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 222–228
April 29 - May 4, 2025

figure represents layer index, from each of which, the y-axis shows the probability (according to logits) of correct target language next token (blue) or English as pivot (orange). While the left figure shows English pivot probability higher than the target token in Greek, such behavior is not observed in the right (Arabic). Inspired, we propose a regulated version, classifying between the two cases, to selectively apply pivoted objective.

Our proposed method, PROM (Pivoted and Regulated Optimization) is shown to be effective on MGSM, XQuAD, MLQA, IndicQA across 9 languages. PROM dominates the baselines in most cases, improving the QA exact match score by 50% overall. The code is publicly available.[1]

2 Pivoted and Regulated Optimization

Preliminaries: Adapting LLM to the Target Language We first formalize the training of LLM architecture as follows:

$$h_0 = f(s), s \in S \quad (1)$$

$$h_i = L_i(h_{i-1}) \quad (2)$$

where L_i is the ith transformer layer in LLM, and f is the embedding layer, S is the set of given inputs. For instruction tuning, typically, only English instruction tuning data sample s_e constructs the input S. The final hidden representation h_N is used for updating the model, where N is the total number of layers.

To enhance the set S for adaptation to the target language, we typically augment each existing English instruction and response $s_e \in S$ with its translated counterpart s_t. Moreover, an additional English to target language translation task sample $s_{e \to t}$ can be added to further align English and the target language (Zhu et al., 2023; Ranaldi et al., 2023; Kuulmets et al., 2024).

2.1 Motivation: Negative Interference

While 'bigger is better' is commonly believed, that adding English instruction tuning samples s_e along with other samples $(s_t, s_{e \to t})$ to construct S is expected to be beneficial (Zhu et al., 2023; Ranaldi et al., 2023), our observation in Section 3.2 indicates the contrary. To explain, we analyze negative interference between two languages, in the latter layers, especially the last layer, which is most relevant to generating the target language (Wendler et al., 2024; Zhao et al., 2024).

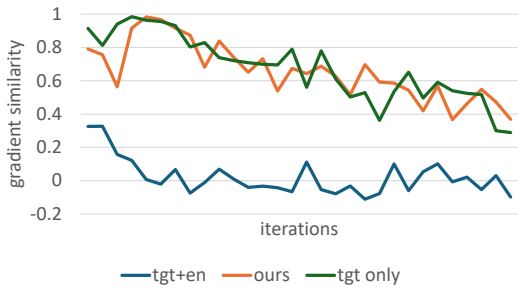

Figure 2: Gradient similarity in the last layer. Lower gradient similarity implies higher negative interference. Ours shows low interference while utilizing both English and the translated data.

Specifically, negative interference (Wang et al., 2020) is quantified using cosine similarity between gradients from two batches composed of different languages (Wang et al., 2020).[2] When such similarity is low, negative interference is considered high, indicating that the gradients are conflicting and pointing in opposite directions.

Figure 2 (blue vs. green) demonstrates that appending English data to the target language results in high negative interference, i.e., low cosine similarity between gradients from two batches. We attribute the suboptimality of appending English data (Section 3.2) to this negative interference.

Our goal is to benefit from English data while avoiding negative interference (orange line in Figure 2). The following subsection introduces how we achieve this.

2.2 Pivoted Objective

We first disentangle the roles of English and target language data. According to Wendler et al. (2024), when generating in non-English using an English-centric LLM, English serves as a pivot language. In other words, forwarding h_n through the LM head for some $n < N$ generates the English version of the next token. This implies that English data is crucial for semantics in the pivot language, while target language data is essential for generating output in the target language.

Next, we devise separate training objectives for each role. To retain semantics while utilizing English data, we design a loss function that considers English as a pivot language. Specifically, we use h_n passed through an LM head for instruction tuning with English data and denote the loss for this as $\mathcal{L}_{n,e}$. Since we are not aiming for exact gen-

[2]Our observation of negative interference is consistently supported in Section 4.

eration, we apply label smoothing with α to $\mathcal{L}_{n,e}$. For language generation using target language data, we use h_N passed through another LM head for instruction tuning and denote the loss for this as $\mathcal{L}_{N,t}$. Finally, we optimize the weighted sum of the two objectives:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{n,e} + \mathcal{L}_{N,t} \quad (3)$$

2.3 Regulated Objective for Under-represented Languages

While the effectiveness of the objective $\mathcal{L}_{n,e}$ depends on the validity of Wendler et al. (2024), recall that their assertion does not apply universally, particularly for under-represented languages in Figure 1, contrasting the scenarios following pivoted assumption (left; Greek) and not (right; Arabic). [3]

We propose to classify such cases by setting $\lambda = 0$ if $\overline{P_{n,e}} < \overline{P_{n,t}}$, where $-$ denotes the average, e denotes English, and t denotes the target language. $P_{n,l}$ denotes the probability of the language l version of the next token in the nth layer, following the definition by Wendler et al. (2024).

3 Experiments

3.1 Experimental Settings

We use LLaMA2-7B (Touvron et al., 2023) as the representative English-centric LLM.

Tasks and Datasets For the English-centric instruction tuning data, we use the ALPACA dataset (Taori et al., 2023). We use Google Translate API to obtain the target language counterpart. For the parallel data for the translation task instruction tuning, we use the WMT23 development dataset,[4] the NTREX (Federmann et al., 2022) and the FLORES (Goyal et al., 2021). We only use these high-quality parallel data, since only high-quality parallel dataset guarantees the performance increase for diverse tasks (Kuulmets et al., 2024).

We evaluate our model on LM-EVALUATION-HARNESS (Gao et al., 2021). We use the available multilingual generative tasks: MSGM (Shi et al., 2023), MLQA (Lewis et al., 2020), and XQuAD (Artetxe et al., 2020). We additionally implement IndicQA (Doddapaneni et al., 2023) evaluation. For QA evaluation, we use the extended version of LM-EVALUATION-HARNESS.[5]

Language Selection Total 9 languages are available in the given datasets:[6] Arabic (ar), Bengali (bn), Greek (el), Malayalam (ml), Marathi (mr), Swahili (sw), Tamil (ta), Telugu (te), and Thai (th).

Implementation Details To perform instruction tuning, we largely follow the setting from Alpaca (Taori et al., 2023).[7] We use learning rate of 2e-5; warmup for 3% of total steps; and train for 3 epochs. We use batch size of 32, sequence length of 1024 or 2048, depending on the GPU consumption. We use $n = 24, \alpha = 0.1, \lambda = 0.1$.[8] Training is done on 8 A100-80GB, taking less than six hours. We evaluate the LLMs with a batch size of 8, in a zero-shot manner. We use the prompts given in the target languages. Evaluation is conducted on an A100, which takes less than two hours.

Comparisons We compare the following methods: a) *LLaMA2*: The baseline English-centric LLM. b) *Bactrian+(t)*: Use the target language data only (Li et al., 2023), enhanced with translation data (Kuulmets et al., 2024), i.e., S consists of $s_t, s_{e\to t}$. c) *xLLAMA2(t+e)*: Add english language data (Zhu et al., 2023), i.e., S consisting of $s_e, s_t, s_{e\to t}$, d) *PROM*: Our proposed method.

3.2 Experimental Results

Negative Interference Drops Performance The final row of Table 3 highlights the positive impact of excluding English instruction tuning data from xLLAMA2(t+e). Across all 11 cases of MGSM and QA evaluation, its exclusion results in superior performance in 8 instances. This supports our claim that naïvely appending translated instruction tuning data incurs negative interference, thereby impairing performance.

Superiority of PROM Tables 1 and 2 show that PROM successfully outperforms the baseline, xLLAMA2(t+e). For example, overall, the exact match score of QA increases by about 50% compared with the baseline. Additionally, as depicted in Table 3, xLLAMA2(t+e) never outperforms PROM, implying PROM is a reliable method for leveraging English instruction tuning data.

Importance of Pivoted Objective The third row in Table 3, identical to the removal of $\mathcal{L}_{n,e}$ entirely, emphasizes the beneficial nature of the proposed $\mathcal{L}_{n,e}$ when contrasted with the first row.

[3] We translated the cloze task in Wendler et al. (2024) for this analysis. We ran in a 5-shot manner. See our results for all languages in Appendix

[4] https://www2.statmt.org/wmt23/translation-task.html

[5] https://github.com/OpenGPTX/lm-evaluation-harness

[6] We use languages whose task performance improves by the baseline adaptation method.

[7] https://github.com/tatsu-lab/stanford_alpaca

[8] We describe the hyperparameter choice in the Appendix.

	XQuAD						MLQA		IndicQA							
	th		ar		el		ar		ta		mr		ml		avg	
	em	f1	em	f1	em	f1	em	f1	em	f1	em	f1	em	f1	em	f1
PROM	**10.7**	**22.5**	3.4*	**16.8***	**5.3**	**22.9**	2.5*	**16.6***	**0.7***	**4.0***	**1.3**	**12.3**	**3.7***	**13.9***	**3.9**	**15.6**
xLLAMA2(t+e)	2.4	14.9	**4.2**	16.6	4.1	20.5	**3.1**	16.4	0.3	3.4	0.3	11.7	3.3	13.1	2.5	13.8
LLaMA2	1.6	9.8	0.1	5.2	1.8	11.4	1.0	7.1	0.0	0.7	0.2	4.3	0.0	0.8	0.7	5.6

Table 1: Exact match and F1 score of diverse QA benchmarks. (*: $\lambda = 0$ for under-represented languages.)

	sw	th	bn	te	avg
PROM	**5.6**	**4.4**	**4.0**	**0.4***	**3.6**
xLLAMA2(t+e)	5.2	4.0	3.2	**0.4**	3.2
LLaMA2	2.4	1.6	0.0	0.0	1.0

Table 2: MGSM Accuracy of comparisons. (*: $\lambda = 0$ for under-represented languages.)

	lose to t+e	wins t+e
PROM	**0/11**	**9/11**
- regulation	2/11	8/11
Bactrian+(t)	1/11	8/11

Table 3: Lose and win counts compared with xLLAMA2(t+e). We deal with 11 QA and MGSM results in Table 1,2. We consider losed or winned if the score of one dominates the other.

Importance of Regulated Objective A comparison between the first and second rows in Table 3 highlights the necessity of regulation.

3.3 Analysis

In this analysis, we show that PROM also deepens the English-pivoting behavior of the LLM. Applying PROM soars up the probability of the English-version of the next token as depicted in the right of Figure 3. This means PROM not only mitigates negative interference, but also improves the pivoting behavior –resulting in a performance increase (Table 1,2).

4 Related Work

Instruction-tuned LLMs for Non-English To extend the capabilities of instruction-tuned LLMs to languages other than English, early attempts involved human annotation of instruction-tuning datasets (Zhang et al., 2023), which lacks scalability.

Wei et al. (2023); Li et al. (2023) leverage LLMs to generate synthetic data for instruction-tuning, however the quality would plummet as the generation ability of LLM for that language decreases

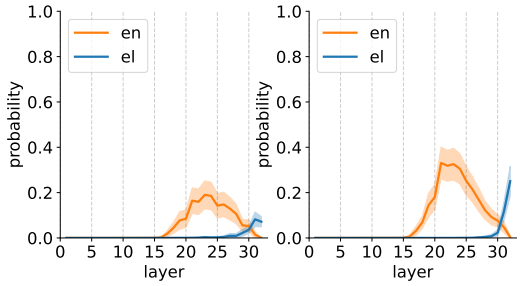

Figure 3: Pivoting behavior (en probability) before (left) and after (right) applying PROM.

than English.

Alternatively, machine-translated instruction-tune datasets (Chen et al., 2023; Holmström and Doostmohammadi, 2023; Santilli and Rodolà, 2023; Cui et al., 2023) paired with higher-quality English instruction-tune data (Zhu et al., 2023; Ranaldi et al., 2023) gained popularity.

Our distinction is observing a possible negative interference between English and target data, and mitigating it by disentangling the roles of the two.

English as a Pivot Language Wendler et al. (2024) explicitly observed pivoting behavior in LLaMA2, an English-centric LLM that the LLM first generates representations for the next token in English at the middle layer before converting them to representations of the target language at the final layer. Our work is inspired by this observation but goes beyond passive observation by (1) recognizing the limitations of their findings for under-represented languages and (2) extending into optimization objectives to mitigate negative interferences.

5 Conclusion

In this paper, we found that appending the English instruction sets along with its translated pairs is not always beneficial, for instruction-tuning in multiple languages. To overcome this, we proposed PROM, where we devised pivoted objective and regulated objective. Experimental results across 9 languages

show the effectiveness of our proposal.

Limitation

We conducted our experiment on only one English-centric LLM, LLaMA2 (Touvron et al., 2023). However, we are following the convention of previous studies (Zhao et al., 2024; Zhu et al., 2023; Kew et al., 2023) that focus on LLaMA for studying English-centric LLMs. We leave applying PROM to other English-centric LLMs, such as Mistral (Jiang et al., 2023), as a future work.

Acknowledgements

This research was partially supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-2020-0-01789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), MSIT/IITP grant (2022-0-00995, 2022-0-00077/RS-2022-II220077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data).

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Peter K Austin and Julia Sallabank. 2011. *The Cambridge Handbook of Endangered Languages*. Cambridge University Press.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Barry Haddow, and Kenneth Heafield. 2023. Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca.

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.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards Leaving No Indic Language Behind: Building Monolingual Corpora, Benchmark and Models for Indic Languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – News Test References for MT Evaluation of 128 Languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation. Zenodo.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *arXiv:2106.03193 [cs]*.

Oskar Holmström and Ehsan Doostmohammadi. 2023. Making Instruction Finetuning Accessible to Non-English Languages: A Case Study on Swedish Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 634–642, Tórshavn, Faroe Islands. University of Tartu Library.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts.

Tannon Kew, Florian Schottmann, and Rico Sennrich. 2023. Turning English-centric LLMs Into Polyglots: How Much Multilinguality Is Needed?

Hele-Andra Kuulmets, Taido Purason, Agnes Luhtaru, and Mark Fishel. 2024. Teaching llama a new language through cross-lingual knowledge transfer.

Patrick Lewis, Barlas Oguz, 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, Online. Association for Computational Linguistics.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-X: Multilingual Replicable Instruction-Following Models with Low-Rank Adaptation.

OpenAI. 2023. GPT-4 Technical Report.

Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2023. Empowering Cross-lingual Abilities of Instruction-tuned Large Language Models by Translation-following demonstrations.

Andrea Santilli and Emanuele Rodolà. 2023. Camoscio: An Italian Instruction-tuned LLaMA.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following LLaMA model.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On Negative Interference in Multilingual Models: Findings and A Meta-Learning Treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023. PolyLM: An Open Source Polyglot Large Language Model.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do Llamas Work in English? On the Latent Language of Multilingual Transformers.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023. BayLing: Bridging Cross-lingual Alignment and Instruction Following through Interactive Translation for Large Language Models.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do Large Language Models Handle Multilingualism?

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating Large Language Models to Non-English by Aligning Languages.

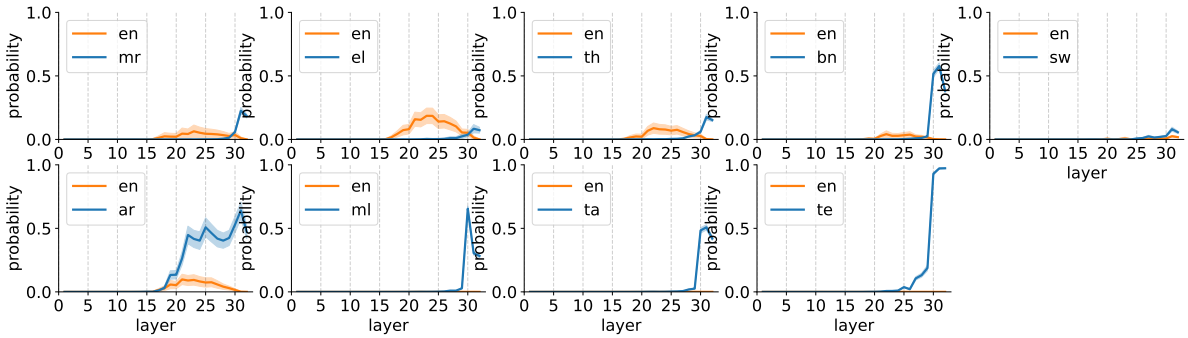

Figure 4: Probability of generating English and the target language tokens per layer.

	n	α	λ	MGSM acc	XQuAD em	XQuAD f1
Bactrian+(t)				4.4	9.0	21.1
ours	24	0.1	0.1	**4.4**	**10.7**	**22.5**
label smooth comparison	24	0.03	0.1	1.6	**11.9**	**24.4**
label smooth comparison	24	0.3	0.1	3.6	**9.4**	**21.5**
label smooth comparison	24	0	0.1	1.2	8.5	19.5
layer id comparison	22	0.1	0.1	1.6	**10.5**	**22.2**
layer id comparison	23	0.1	0.1	2.4	8.3	19.2
layer id comparison	25	0.1	0.1	2.4	**9.2**	20.6
layer id comparison	26	0.1	0.1	2.8	6.9	18.3
loss weight comparision	24	0.1	0.3	1.2	7.9	19.3
loss weight comparision	24	0.1	0.5	2.8	8.2	**21.1**
loss weight comparision	24	0.1	1	3.6	4.0	17.2

Table 4: Comparison on thai (th) language varying hyperparameters.

A Appendix

A.1 Full Results for Figure 1

Figure 4 reports our results for nine languages, with and without pivoted behaviors.

A.2 The Choice of Hyperparameters

We tuned N, α, λ on thai language as Table 4. Only our setting is on par or outperform the best baseline, Bactrian+(t). Note that removing the thai columns from Table 1, 2 does not change the trend or analysis.