Let's Think 一步一步: A Cognitive Framework for Characterizing Code-Switching in LLM Reasoning

Eleanor M. Lin
EECS
University of Michigan
elealin@umich.edu

David Jurgens
EECS, SI
University of Michigan
jurgens@umich.edu

Abstract

State-of-the-art large language models (LLMs) code-switch (i.e., mix languages), but how and why is still poorly understood–especially cognitive differences from humans. We address this gap by introducing a cognitive framework for characterizing code-switching in LLM reasoning. We start from reasoning examples sourced from diverse models, languages, domains, and tasks. Fusing top-down theory-driven and bottom-up data-driven approaches, we then develop a taxonomy of code-switched reasoning behaviors. Our taxonomy reveals that LLM and human code-switching behaviors in LLMs partially align. Additionally, more naturalistic, human-like code-switching may boost model performance, particularly for languages from the long tail of training data distributions. Our work serves as a first, necessary step toward uncovering parallels between LLM and human code-switching. With further testing, LLMs could potentially serve as proxies for human multilingual cognition. Additionally, our approach can develop future reasoning taxonomies informed by cognitive science and education.

1 Introduction

Code-switching occurs when multiple languages are integrated into a single communication [Myers-Scotton, 2017]. Work in both linguistics and natural language processing suggests that code-switching may facilitate reasoning, in both humans and LLMs (e.g., DeepSeek-AI et al. [2025], Torregrossa et al. [2025], and Li et al. [2025]). Here, we aim to better understand how LLMs code-switch when reasoning, asking the following questions:

- RQ 1 How do LLMs code-switch during reasoning?
- RQ 2 How does code-switching in LLM reasoning parallel and differ from code-switching in humans?
- RQ 3 Where does code-switching in reasoning help performance on tasks requiring reasoning?

To address this gap, we make three main contributions:

(1) We introduce a large-scale dataset of multilingual LLM reasoning traces specifically designed to allow the study of code-switched LLM reasoning, covering reasoning from diverse models, domains, tasks, and languages.

(2) We use our dataset to develop a cognitive framework for characterizing code-switched reasoning behaviors (see Figure 1).

(3) We apply our framework to understand how models use code-switching to perform multilingual reasoning.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models.

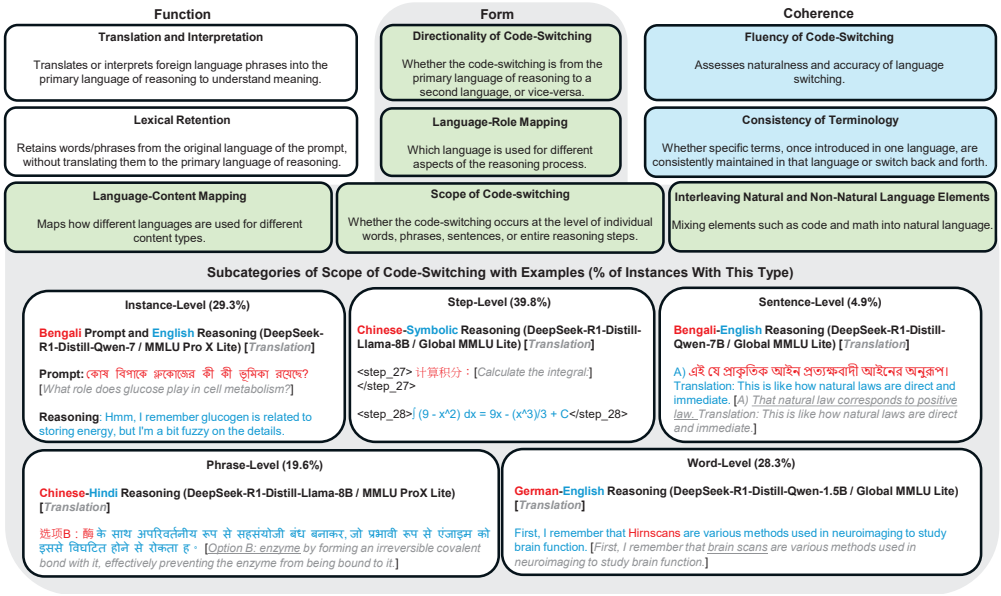

Figure 1: Selected code-switched reasoning behaviors from our taxonomy. Percentages of reasoning examples for the subcategories of the **Scope of Code-switching** category add to over 100% because a single reasoning instance can potentially feature code-switching at multiple scopes.

We find that:

(RQ 1) LLM code-switching serves diverse functions, including translating the prompt from its original language to another language within the model reasoning. LLMs also use the opposite strategy of retaining terms from the prompt in their original language, within reasoning that is primarily in a different language.

(RQ 2) Code-switching in LLMs partially aligns with human code-switching behaviors. For example, LLM handling of languages in the long tail of the training distribution follows similar patterns to compensatory code-switching by human bilinguals who have uneven proficiency in two languages.

(RQ 3) More naturalistic, human-like code-switching may improve generalization to languages underrepresented in the training data.

By investigating how LLM code-switching parallels and diverges from human code-switching, our work contributes to growing research on comparing and modeling human cognition with LLMs [Binz and Schulz, 2024, Ji-An et al., 2024, Tan et al., 2024]. Insights gleaned from applying our taxonomy can be also applied to develop reasoning systems that integrate code-switching as a core feature. Code and data are available at `https://figshare.com/s/f8f6fd2d899b93077b03`.

2 Related work

We contribute to growing work on controlling reasoning language (e.g., Tam et al. [2025b] and Gao et al. [2025]). Here, we discuss only work that characterizes model reasoning in a principled way.

Prior work on code-switched reasoning lacks our coverage of diverse models, languages, domains, tasks, and reasoning behaviors. Yong et al. [2025] study the English-centric s1 model family using commonsense, factual, cultural, and causal reasoning benchmarks, with prompts in four languages other than English. Li et al. [2025] study Chinese/English math reasoning from the Chinese-English QwQ32B-Preview model. Wang et al. [2025] study the Chinese-English DeepSeek-R1 model family and multilingual QwQ-32B, Qwen3, and Gemini 2.0 Flash Thinking models on commonsense, factual, and logical reasoning in 15 languages. In contrast, we cover 15 models from 9 families with diverse multilingual capabilities, 18 prompt languages, and reasoning domains/tasks beyond those in prior work, e.g., moral reasoning. When characterizing code-switching behaviors, Yong et al. identify the "quote-and-think" pattern and apply the pre-existing linguistic concepts of a

matrix language and intra-/inter-sentential switching. Wang et al. quantify relative proportions of reasoning languages. Li et al. identify four main patterns. In contrast, we introduce a rich taxonomy of 17 code-switching patterns across three dimensions and five levels of granularity.

Our work combines top-down, theory-driven and bottom-up, data-driven approaches for classifying reasoning behavior. Gandhi et al. [2025] demonstrated the value of theory-driven approaches by identifying reasoning model behaviors that both drive performance and parallel human behaviors. In contrast, CoT Encyclopedia characterizes reasoning using bottom-up LLM-assisted brainstorming and clustering of the resulting text [Lee et al., 2025]. While we base our own approach on that of Lee et al., we focus specifically on code-switching. We introduce manual curation as an extra step to ground the resulting taxonomy of code-switching behaviors in prior real-world observations and theories of code-switching. Additionally, while Lee et al. classify entire reasoning instances, our framework enables finer-grained analysis of reasoning, at the instance, step, sentence, and intra-sentence levels. Finally, CoT Encyclopedia has not been tested on smaller multilingual models and domains beyond math, science, and general knowledge, whereas we cover diverse domains and models.

3 Approach

Both top-down and bottom-up approaches to characterizing reasoning behaviors (see section 2) face limitations. Theory-grounded, top-down approaches which look for pre-defined, human-aligned behaviors in model reasoning (e.g., [Gandhi et al., 2025]) may miss novel reasoning behaviors that do not align with humans. On the other hand, data-driven, bottom-up approaches which derive categories of behaviors from observations of model reasoning (e.g., [Lee et al., 2025]) may fail to surface misalignment with categories of human behaviors. In particular, these data-driven, bottom-up approaches may fail to observe behaviors that only appear in humans, and not in models. To address these limtiations, we fuse top-down theory-driven and bottom-up data-driven approaches in our framework.

First, we select data from diverse models, languages, tasks, and domains for generalizability. We source examples from 15 models of diverse sizes, reasoning capabilities, and multilingual capabilities. We select prompts for generating reasoning examples from seven datasets, from s1K (covering STEM, law, logic, puzzles, and humanities) to UniMoral (covering moral reasoning). Prompts cover 18 languages (Arabic, Bengali, Burmese, English, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Mandarin Chinese, Portuguese, Russian, Spanish, Swahili, Thai, and Yoruba), 10 scripts, and 8 language families [Hammarström et al., 2025, Unicode, 2025]. We include about 50 instances from each model/language/dataset combination, for about 7,000 instances total. See Appendices A.3-A.5, A.25, and A.26 for full details on models, datasets, and prompting configurations.

Next, we develop a taxonomy of code-switched reasoning behaviors by fusing top-down theory-driven and bottom-up data-driven approaches. We rely on the reasoning examples described above as input context to Gemini 2.5 Flash for brainstorming criteria that differentiate code-switching strategies [Google, 2025a, Gemini Team et al., 2025]. At this stage, we introduce helpful inductive biases to guide the taxonomy development by providing Gemini 2.5 Flash with a minimal and broad definition of code-switching ("use of multiple scripts or languages"). We apply topic modeling to Gemini's brainstormed criteria, using the BERTopic pipeline [Grootendorst, 2022]. Finally, we introduce further helpful structure in a top-down manner. In particular, we manually consolidate redundant topics, dropping those unlikely to generalize (e.g., "Interpretation of Yoruba Terms"), and group topics into dimensions, categories, and subcategories. Manual curation of topics into the final hierarchy is guided by the authors' knowledge of the relevant literature on code-switching and reasoning in both humans and LLMs. See Appendices A.7 and A.8 for details.

Finally, we annotate our taxonomy on previously unseen reasoning examples. From the datasets listed in Appendix A.26 (but excluding any instances already used to develop the taxonomy), we include about 50 instances from each model/dataset/language combination. We prompt Gemini 2.5 Flash to annotate each instance with our taxonomy categories. We use newlines to identify reasoning step boundaries as in Chen et al. [2025]. See Appendices A.5 and A.9-A.19 for details.

We validate LLM annotations by human evaluation. Our taxonomy development approach follows Lee et al. [2025], who conduct extensive human evaluation demonstrating that their CoT ENCYLOPEDIA derives more interpretable, comprehensive analyses than prior approaches. To further

verify the quality of the LLM annotations used in this work, we follow a similar procedure to Lee et al. We randomly sample 100 instances each from (1) the code-switching criteria brainstormed by Gemini 2.5 Flash, and (2) the reasoning examples annotated with our taxonomy, then manually annotate them as detailed in Appendix A.20.

4 Results

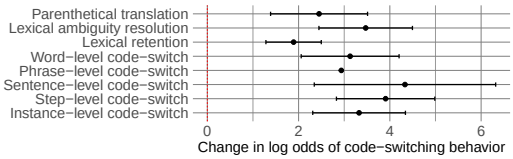

Figure 2: When prompted in a language they do not officially support, models are significantly more likely to engage in diverse code-switching behaviors during reasoning, as demonstrated for the selected categories from our taxonomy. Error bars show standard error. All effects are significant at the $\alpha = 0.001$ level (see Appendix A.22 for details on modeling for this figure and Figure 3).

Our taxonomy (see Figure 1) features 3 dimensions (Function, Form, and Coherence) annotated at the word, phrase, sentence, step, instance, and code-switch–level. Behaviors under each dimension are classified hierarchically. (See Appendix A.6 for the detailed taxonomy.) We find that Gemini 2.5 Flash is an effective annotator, generating plausible descriptions of code-switching 85% of the time.

In answer to our research questions, we find that (1) models are predisposed to use diverse code-switching behaviors to handle queries in languages in the long tail of their training data distributions (**RQ 1**), (2) model behavior partially aligns with humans (**RQ 2**), and (3) code-switching fluency positively affects performance, with an even stronger effect when prompting in unsupported languages (**RQ 3**).

Gemini 2.5 Flash performs reasonably well at annotating code-switched reasoning. For 85% of reasoning instances, the criteria brainstormed by Gemini 2.5 Flash for differentiating code-switching strategies are plausible. When annotating taxonomy categories, Gemini 2.5 Flash achieves an average macro F1 of 0.60 for the binary labels evaluated, outperforming a naive majority class baseline (macro F1 $= 0.46$). For code-switching fluency (a continuous measure), Gemini 2.5 Flash and human ratings are weakly positively correlated ($r = 0.29, p < 0.01$). The low correlation between LLM and human fluency ratings may be partially due to the subjective nature of this rating, where the rater is asked to rate fluency on a scale from one to five.

RQ 1: Models are more likely to code-switch to handle queries in unsupported languages. For the DeepSeek distilled, Llama-3.1-Nemotron-Nano-8B-v1, and Phi-4-mini-reasoning models, prompting in an unsupported language significantly increases the log odds that the model engages in diverse code-switching behaviors during reasoning, as seen in Figure 2. (See Appendix A.21 for languages supported by each model.) In particular, applying our taxonomy reveals that code-switches at all levels (**word**, **phrase**, **sentence**, **step**, and **instance**) are more likely when prompted in an unsupported language. Additionally, behaviors from the **Translation and Interpretation** and **Lexical Retention** categories of our taxonomy's **Function** dimension are more likely when prompting in unsupported languages.

Overall, this result suggests that diverse forms of code-switching are a generalizable behavior across reasoning models for handling languages beyond the training distribution. In this respect, model behavior parallels compensatory code-switching in humans, which occurs when a multilingual switches to one language to compensate for a lack of proficiency in another language. For example, a German-Turkish bilingual may say a word in German because they are able to think of this word more easily in German than in Turkish [Schächinger Tenés et al., 2023].

RQ 2: Model code-switching behavior partially aligns with humans. For example, models may quote content in one language and then translate and reason about that content in English. Our taxonomy categories of **Translation and Interpretation** and **Language of Direct Quotes** effectively describe this behavior. Similarly, humans may quote speech in its original language, but provide additional context for the quote in another language [Begum et al., 2016]. Humans also code-switch to repeat phrases from one language in another language, for both emphasis and clarity [Belani and Flanigan, 2023]. See Appendix A.27 for examples.

By code-switching to reason about content from lower-resource languages in higher-resource languages, models mitigate the language resource gap. In Figure 1, we show additional examples of how code-switching at the **instance**, **step**, **sentence**, **phrase**, and **word levels** of our taxonomy's **Scope of Code-Switching** category handles queries in languages (Bengali, Hindi, and German) beyond the core languages (Chinese and English) supported by the DeepSeek-R1 distilled models.

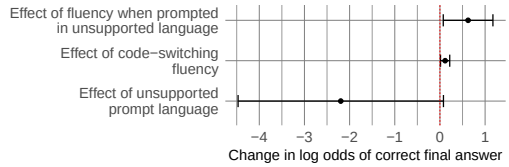

Figure 3: The **Fluency** category of the **Coherence** dimension in our taxonomy is defined as the accuracy and naturalness of code-switching. Increased fluency increases the log odds of a correct final answer (considering code-switched instances only). Error bars show 95% CI. Fluency effect significant at the $\alpha = 0.01$ level.

RQ 3: More fluent code-switching has a significant positive effect on performance. In our taxonomy, the **Fluency**[1] category of the **Coherence** dimension is defined as the accuracy and naturalness of the code-switching, rated on a scale from one ("very disfluent") to five ("very fluent"). We measure the effect of code-switching on performance using a generalized linear mixed effects (GLME) model, with correctness of the final LLM answer coded as a binary variable. Our GLME model includes the interaction between fluency and whether or not the prompt language is supported by the model, as well as the random effects of the dataset item and model (see Appendix A.22 for details).

As seen in Figure 3, for each one-point increase in the fluency rating of a code-switched reasoning instance, the log odds of a correct final answer increase by 0.12 ($p < 0.01$). The positive effect of code-switching fluency on accuracy suggests that code-switching that is more natural and human-like may be more beneficial to model performance. Moreover, we see that increased code-switching fluency has an even greater positive effect (0.62 ± 0.28 standard errors, $p < 0.05$) when models are prompted in unsupported languages. In other words, while beneficial for model performance on average, fluent code-switching is particularly impactful when models are forced to handle queries in languages that are underrepresented in their training data. Our finding in LLMs parallels the finding that highly fluent human code-switchers demonstrate cognitive benefits, compared to less fluent bilinguals [Kheder and Kaan, 2021].

5 Conclusion

In this work, we ask (1) how LLMs code-switch during reasoning, (2) how this code-switching parallels and differs from human code-switching, and (3) where code-switching can help performance on reasoning tasks. To answer these questions, we introduce a new dataset of LLM reasoning traces for studying code-switching and a cognitive framework for characterizing model code-switched reasoning behaviors. The dataset consists of reasoning examples from diverse domains, tasks, and models. We use this dataset in our human-validated approach, combining LLM-assisted brainstorming, topic modeling, and theory-informed manual curation to develop a taxonomy of behaviors. In answer to our research questions, we find that (1) models code-switch to reason about queries in languages from the long tail of their training data distributions, (2) humans and model behaviors partially align, and (3) fluent code-switching during reasoning positively impacts performance, suggesting that more natural and human-like code-switching may benefit reasoning. This beneficial effect is even more pronounced in the challenging setting of prompting in unsupported languages. Overall, the surface-level parallels between LLM and human code-switchers suggest that with further testing, LLMs could serve as proxies for human multilingual cognition in the future. Our combination of theory- and data-driven approaches for characterizing model behavior can also extend to behaviors beyond code-switching. See Appendices A.1 and A.2 for future work, limitations, and broader impacts.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Award No. DGE 2241144.

[1]This definition of "fluency" is specific to our taxonomy and differs from the general understanding of fluency as describing a speaker's proficiency in a particular language.

References

Anthropic. Claude 3.7 sonnet and claude code, 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet.

R. Begum, K. Bali, M. Choudhury, K. Rudra, and N. Ganguly. Functions of code-switching in tweets: An annotation framework and some initial experiments. In N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1644–1650, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1260/.

R. Belani and J. Flanigan. Automatic identification of code-switching functions in speech transcripts. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7438–7448, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.469. URL https://aclanthology.org/2023.findings-acl.469/.

A. Bercovich, I. Levy, I. Golan, M. Dabbah, R. El-Yaniv, O. Puny, I. Galil, Z. Moshe, T. Ronen, N. Nabwani, I. Shahaf, O. Tropp, E. Karpas, R. Zilberstein, J. Zeng, S. Singhal, A. Bukharin, Y. Zhang, T. Konuk, G. Shen, A. S. Mahabaleshwarkar, B. Kartal, Y. Suhara, O. Delalleau, Z. Chen, Z. Wang, D. Mosallanezhad, A. Renduchintala, H. Qian, D. Rekesh, F. Jia, S. Majumdar, V. Noroozi, W. U. Ahmad, S. Narenthiran, A. Ficek, M. Samadi, J. Huang, S. Jain, I. Gitman, I. Moshkov, W. Du, S. Toshniwal, G. Armstrong, B. Kisacanin, M. Novikov, D. Gitman, E. Bakhturina, P. Varshney, M. Narsimhan, J. P. Scowcroft, J. Kamalu, D. Su, K. Kong, M. Kliegl, R. Karimi, Y. Lin, S. Satheesh, J. Parmar, P. Gundecha, B. Norick, J. Jennings, S. Prabhumoye, S. N. Akter, M. Patwary, A. Khattar, D. Narayanan, R. Waleffe, J. Zhang, B.-Y. Su, G. Huang, T. Kong, P. Chadha, S. Jain, C. Harvey, E. Segal, J. Huang, S. Kashirsky, R. McQueen, I. Putterman, G. Lam, A. Venkatesan, S. Wu, V. Nguyen, M. Kilaru, A. Wang, A. Warno, A. Somasamudramath, S. Bhaskar, M. Dong, N. Assaf, S. Mor, O. U. Argov, S. Junkin, O. Romanenko, P. Larroy, M. Katariya, M. Rovinelli, V. Balas, N. Edelman, A. Bhiwandiwalla, M. Subramaniam, S. Ithape, K. Ramamoorthy, Y. Wu, S. V. Velury, O. Almog, J. Daw, D. Fridman, E. Galinkin, M. Evans, S. Ghosh, K. Luna, L. Derczynski, N. Pope, E. Long, S. Schneider, G. Siman, T. Grzegorzek, P. Ribalta, M. Katariya, C. Alexiuk, J. Conway, T. Saar, A. Guan, K. Pawelec, S. Prayaga, O. Kuchaiev, B. Ginsburg, O. Olabiyi, K. Briski, J. Cohen, B. Catanzaro, J. Alben, Y. Geifman, and E. Chung. Llama-nemotron: Efficient reasoning models, 2025. URL https://arxiv.org/abs/2505.00949.

M. Binz and E. Schulz. Turning large language models into cognitive models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eiC4BKypf1.

R. Chen, Z. Zhang, J. Hong, S. Kundu, and Z. Wang. Seal: Steerable reasoning calibration of large language models for free, 2025. URL https://arxiv.org/abs/2504.07986.

DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

M. Fernandez-Duque. *The Effects of Dual-Language Experience on Memory and Creativity*. PhD thesis, Northwestern University, 2025. URL https://proxy.lib.umich.edu/login?url=https://www.proquest.com/dissertations-theses/effects-dual-language-experience-on-memory/docview/3215610716/se-2. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2025-06-26.

K. Gandhi, A. Chakravarthy, A. Singh, N. Lile, and N. D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

C. Gao, X. Huang, W. Zhu, S. Huang, L. Li, and F. Yuan. Could thinking multilingually empower llm reasoning?, 2025. URL https://arxiv.org/abs/2504.11833.

L. Gao, J. Tow, B. Abbasi, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, A. Le Noac'h, H. Li, K. McDonell, N. Muennighoff, C. Ociepa, J. Phang, L. Reynolds, H. Schoelkopf, A. Skowron, L. Sutawika, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gemini Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, D. Silver, M. Johnson, I. Antonoglou, J. Schrittwieser, A. Glaese, J. Chen, E. Pitler, T. Lillicrap, A. Lazaridou, O. Firat, J. Molloy, M. Isard, P. R. Barham, T. Hennigan, B. Lee, F. Viola, M. Reynolds, Y. Xu, R. Doherty, E. Collins, C. Meyer, E. Rutherford, E. Moreira, K. Ayoub, M. Goel, J. Krawczyk, C. Du, E. Chi, H.-T. Cheng, E. Ni, P. Shah, P. Kane, B. Chan, M. Faruqui, A. Severyn, H. Lin, Y. Li, Y. Cheng, A. Ittycheriah, M. Mahdieh, M. Chen, P. Sun, D. Tran, S. Bagri, B. Lakshminarayanan, J. Liu, A. Orban, F. Güra, H. Zhou, X. Song, A. Boffy, H. Ganapathy, S. Zheng, H. Choe, Ágoston Weisz, T. Zhu, Y. Lu, S. Gopal, J. Kahn, M. Kula, J. Pitman, R. Shah, E. Taropa, M. A. Merey, M. Baeuml, Z. Chen, L. E. Shafey, Y. Zhang, O. Sercinoglu, G. Tucker, E. Piqueras, M. Krikun, I. Barr, N. Savinov, I. Danihelka, B. Roelofs, A. White, A. Andreassen, T. von Glehn, L. Yagati, M. Kazemi, L. Gonzalez, M. Khalman, J. Sygnowski, A. Frechette, C. Smith, L. Culp, L. Proleev, Y. Luan, X. Chen, J. Lottes, N. Schucher, F. Lebron, A. Rrustemi, N. Clay, P. Crone, T. Kocisky, J. Zhao, B. Perz, D. Yu, H. Howard, A. Bloniarz, J. W. Rae, H. Lu, L. Sifre, M. Maggioni, F. Alcober, D. Garrette, M. Barnes, S. Thakoor, J. Austin, G. Barth-Maron, W. Wong, R. Joshi, R. Chaabouni, D. Fatiha, A. Ahuja, G. S. Tomar, E. Senter, M. Chadwick, I. Kornakov, N. Attaluri, I. Iturrate, R. Liu, Y. Li, S. Cogan, J. Chen, C. Jia, C. Gu, Q. Zhang, J. Grimstad, A. J. Hartman, X. Garcia, T. S. Pillai, J. Devlin, M. Laskin, D. de Las Casas, D. Valter, C. Tao, L. Blanco, A. P. Badia, D. Reitter, M. Chen, J. Brennan, C. Rivera, S. Brin, S. Iqbal, G. Surita, J. Labanowski, A. Rao, S. Winkler, E. Parisotto, Y. Gu, K. Olszewska, R. Addanki, A. Miech, A. Louis, D. Teplyashin, G. Brown, E. Catt, J. Balaguer, J. Xiang, P. Wang, Z. Ashwood, A. Briukhov, A. Webson, S. Ganapathy, S. Sanghavi, A. Kannan, M.-W. Chang, A. Stjerngren, J. Djolonga, Y. Sun, A. Bapna, M. Aitchison, P. Pejman, H. Michalewski, T. Yu, C. Wang, J. Love, J. Ahn, D. Bloxwich, K. Han, P. Humphreys, T. Sellam, J. Bradbury, V. Godbole, S. Samangooei, B. Damoc, A. Kaskasoli, S. M. R. Arnold, V. Vasudevan, S. Agrawal, J. Riesa, D. Lepikhin, R. Tanburn, S. Srinivasan, H. Lim, S. Hodkinson, P. Shyam, J. Ferret, S. Hand, A. Garg, T. L. Paine, J. Li, Y. Li, M. Giang, A. Neitz, Z. Abbas, S. York, M. Reid, E. Cole, A. Chowdhery, D. Das, D. Rogozińska, V. Nikolaev, P. Sprechmann, Z. Nado, L. Zilka, F. Prost, L. He, M. Monteiro, G. Mishra, C. Welty, J. Newlan, D. Jia, M. Allamanis, C. H. Hu, R. de Liedekerke, J. Gilmer, C. Saroufim, S. Rijhwani, S. Hou, D. Shrivastava, A. Baddepudi, A. Goldin, A. Ozturel, A. Cassirer, Y. Xu, D. Sohn, D. Sachan, R. K. Amplayo, C. Swanson, D. Petrova, S. Narayan, A. Guez, S. Brahma, J. Landon, M. Patel, R. Zhao, K. Villela, L. Wang, W. Jia, M. Rahtz, M. Giménez, L. Yeung, J. Keeling, P. Georgiev, D. Mincu, B. Wu, S. Haykal, R. Saputro, K. Vodrahalli, J. Qin, Z. Cankara, A. Sharma, N. Fernando, W. Hawkins, B. Neyshabur, S. Kim, A. Hutter, P. Agrawal, A. Castro-Ros, G. van den Driessche, T. Wang, F. Yang, S. yiin Chang, P. Komarek, R. McIlroy, M. Lučić, G. Zhang, W. Farhan, M. Sharman, P. Natsev, P. Michel, Y. Bansal, S. Qiao, K. Cao, S. Shakeri, C. Butterfield, J. Chung, P. K. Rubenstein, S. Agrawal, A. Mensch, K. Soparkar, K. Lenc, T. Chung, A. Pope, L. Maggiore, J. Kay, P. Jhakra, S. Wang, J. Maynez, M. Phuong, T. Tobin, A. Tacchetti, M. Trebacz, K. Robinson, Y. Katariya, S. Riedel, P. Bailey, K. Xiao, N. Ghelani, L. Aroyo, A. Slone, N. Houlsby, X. Xiong, Z. Yang, E. Gribovskaya, J. Adler, M. Wirth, L. Lee, M. Li, T. Kagohara, J. Pavagadhi, S. Bridgers, A. Bortsova, S. Ghemawat, Z. Ahmed, T. Liu, R. Powell, V. Bolina, M. Iinuma, P. Zablotskaia, J. Besley, D.-W. Chung, T. Dozat, R. Comanescu, X. Si, J. Greer, G. Su, M. Polacek, R. L. Kaufman, S. Tokumine, H. Hu, E. Buchatskaya, Y. Miao, M. Elhawaty, A. Siddhant, N. Tomasev, J. Xing, C. Greer, H. Miller, S. Ashraf, A. Roy, Z. Zhang, A. Ma, A. Filos, M. Besta, R. Blevins, T. Klimenko, C.-K. Yeh, S. Changpinyo, J. Mu, O. Chang, M. Pajarskas, C. Muir, V. Cohen, C. L. Lan, K. Haridasan, A. Marathe, S. Hansen, S. Douglas, R. Samuel, M. Wang, S. Austin, C. Lan, J. Jiang, J. Chiu, J. A. Lorenzo, L. L. Sjösund, S. Cevey, Z. Gleicher, T. Avrahami, A. Boral, H. Srinivasan, V. Selo, R. May, K. Aisopos, L. Hussenot, L. B. Soares, K. Baumli, M. B. Chang, A. Recasens, B. Caine, A. Pritzel, F. Pavetic, F. Pardo, A. Gergely, J. Frye, V. Ramasesh, D. Horgan, K. Badola, N. Kassner, S. Roy, E. Dyer, V. C. Campos, A. Tomala, Y. Tang, D. E. Badawy, E. White, B. Mustafa, O. Lang, A. Jindal, S. Vikram, Z. Gong, S. Caelles, R. Hemsley, G. Thornton, F. Feng, W. Stokowiec, C. Zheng, P. Thacker, Çağlar Ünlü, Z. Zhang, M. Saleh, J. Svensson, M. Bileschi, P. Patil, A. Anand, R. Ring, K. Tsihlas, A. Vezer, M. Selvi, T. Shevlane, M. Rodriguez, T. Kwiatkowski, S. Daruki, K. Rong, A. Dafoe, N. FitzGerald, K. Gu-Lemberg, M. Khan, L. A. Hendricks, M. Pellat, V. Feinberg, J. Cobon-Kerr, T. Sainath, M. Rauh, S. H. Hashemi, R. Ives, Y. Hasson, E. Noland, Y. Cao, N. Byrd, L. Hou, Q. Wang, T. Sottiaux, M. Paganini, J.-B. Lespiau, A. Moufarek, S. Hassan, K. Shivakumar, J. van Amersfoort, A. Mandhane, P. Joshi, A. Goyal, M. Tung, A. Brock, H. Sheahan, V. Misra, C. Li, N. Rakićević, M. Dehghani, F. Liu, S. Mittal, J. Oh, S. Noury, E. Sezener, F. Huot, M. Lamm, N. D. Cao, C. Chen, S. Mudgal, R. Stella,

K. Brooks, G. Vasudevan, C. Liu, M. Chain, N. Melinkeri, A. Cohen, V. Wang, K. Seymore, S. Zubkov, R. Goel, S. Yue, S. Krishnakumaran, B. Albert, N. Hurley, M. Sano, A. Mohananey, J. Joughin, E. Filonov, T. Kępa, Y. Eldawy, J. Lim, R. Rishi, S. Badiezadegan, T. Bos, J. Chang, S. Jain, S. G. S. Padmanabhan, S. Puttagunta, K. Krishna, L. Baker, N. Kalb, V. Bedapudi, A. Kurzrok, S. Lei, A. Yu, O. Litvin, X. Zhou, Z. Wu, S. Sobell, A. Siciliano, A. Papir, R. Neale, J. Bragagnolo, T. Toor, T. Chen, V. Anklin, F. Wang, R. Feng, M. Gholami, K. Ling, L. Liu, J. Walter, H. Moghaddam, A. Kishore, J. Adamek, T. Mercado, J. Mallinson, S. Wandekar, S. Cagle, E. Ofek, G. Garrido, C. Lombriser, M. Mukha, B. Sun, H. R. Mohammad, J. Matak, Y. Qian, V. Peswani, P. Janus, Q. Yuan, L. Schelin, O. David, A. Garg, Y. He, O. Duzhyi, A. Älgmyr, T. Lottaz, Q. Li, V. Yadav, L. Xu, A. Chinien, R. Shivanna, A. Chuklin, J. Li, C. Spadine, T. Wolfe, K. Mohamed, S. Das, Z. Dai, K. He, D. von Dincklage, S. Upadhyay, A. Maurya, L. Chi, S. Krause, K. Salama, P. G. Rabinovitch, P. K. R. M, A. Selvan, M. Dektiarev, G. Ghiasi, E. Guven, H. Gupta, B. Liu, D. Sharma, I. H. Shtacher, S. Paul, O. Akerlund, F.-X. Aubet, T. Huang, C. Zhu, E. Zhu, E. Teixeira, M. Fritze, F. Bertolini, L.-E. Marinescu, M. Bölle, D. Paulus, K. Gupta, T. Latkar, M. Chang, J. Sanders, R. Wilson, X. Wu, Y.-X. Tan, L. N. Thiet, T. Doshi, S. Lall, S. Mishra, W. Chen, T. Luong, S. Benjamin, J. Lee, E. Andrejczuk, D. Rabiej, V. Ranjan, K. Styrc, P. Yin, J. Simon, M. R. Harriott, M. Bansal, A. Robsky, G. Bacon, D. Greene, D. Mirylenka, C. Zhou, O. Sarvana, A. Goyal, S. Andermatt, P. Siegler, B. Horn, A. Israel, F. Pongetti, C.-W. L. Chen, M. Selvatici, P. Silva, K. Wang, J. Tolins, K. Guu, R. Yogev, X. Cai, A. Agostini, M. Shah, H. Nguyen, N. O. Donnaile, S. Pereira, L. Friso, A. Stambler, A. Kurzrok, C. Kuang, Y. Romanikhin, M. Geller, Z. Yan, K. Jang, C.-C. Lee, W. Fica, E. Malmi, Q. Tan, D. Banica, D. Balle, R. Pham, Y. Huang, D. Avram, H. Shi, J. Singh, C. Hidey, N. Ahuja, P. Saxena, D. Dooley, S. P. Potharaju, E. O'Neill, A. Gokulchandran, R. Foley, K. Zhao, M. Dusenberry, Y. Liu, P. Mehta, R. Kotikalapudi, C. Safranek-Shrader, A. Goodman, J. Kessinger, E. Globen, P. Kolhar, C. Gorgolewski, A. Ibrahim, Y. Song, A. Eichenbaum, T. Brovelli, S. Potluri, P. Lahoti, C. Baetu, A. Ghorbani, C. Chen, A. Crawford, S. Pal, M. Sridhar, P. Gurita, A. Mujika, I. Petrovski, P.-L. Cedoz, C. Li, S. Chen, N. D. Santo, S. Goyal, J. Punjabi, K. Kappaganthu, C. Kwak, P. LV, S. Velury, H. Choudhury, J. Hall, P. Shah, R. Figueira, M. Thomas, M. Lu, T. Zhou, C. Kumar, T. Jurdi, S. Chikkerur, Y. Ma, A. Yu, S. Kwak, V. Ähdel, S. Rajayogam, T. Choma, F. Liu, A. Barua, C. Ji, J. H. Park, V. Hellendoorn, A. Bailey, T. Bilal, H. Zhou, M. Khatir, C. Sutton, W. Rzadkowski, F. Macintosh, R. Vij, K. Shagin, P. Medina, C. Liang, J. Zhou, P. Shah, Y. Bi, A. Dankovics, S. Banga, S. Lehmann, M. Bredesen, Z. Lin, J. E. Hoffmann, J. Lai, R. Chung, K. Yang, N. Balani, A. Bražinskas, A. Sozanschi, M. Hayes, H. F. Alcalde, P. Makarov, W. Chen, A. Stella, L. Snijders, M. Mandl, A. Kärrman, P. Nowak, X. Wu, A. Dyck, K. Vaidyanathan, R. R, J. Mallet, M. Rudominer, E. Johnston, S. Mittal, A. Udathu, J. Christensen, V. Verma, Z. Irving, A. Santucci, G. Elsayed, E. Davoodi, M. Georgiev, I. Tenney, N. Hua, G. Cideron, E. Leurent, M. Alnahlawi, I. Georgescu, N. Wei, I. Zheng, D. Scandinaro, H. Jiang, J. Snoek, M. Sundararajan, X. Wang, Z. Ontiveros, I. Karo, J. Cole, V. Rajashekhar, L. Tumeh, E. Ben-David, R. Jain, J. Uesato, R. Datta, O. Bunyan, S. Wu, J. Zhang, P. Stanczyk, Y. Zhang, D. Steiner, S. Naskar, M. Azzam, M. Johnson, A. Paszke, C.-C. Chiu, J. S. Elias, A. Mohiuddin, F. Muhammad, J. Miao, A. Lee, N. Vieillard, J. Park, J. Zhang, J. Stanway, D. Garmon, A. Karmarkar, Z. Dong, J. Lee, A. Kumar, L. Zhou, J. Evens, W. Isaac, G. Irving, E. Loper, M. Fink, I. Arkatkar, N. Chen, I. Shafran, I. Petrychenko, Z. Chen, J. Jia, A. Levskaya, Z. Zhu, P. Grabowski, Y. Mao, A. Magni, K. Yao, J. Snaider, N. Casagrande, E. Palmer, P. Suganthan, A. Castaño, I. Giannoumis, W. Kim, M. Rybiński, A. Sreevatsa, J. Prendki, D. Soergel, A. Goedeckemeyer, W. Gierke, M. Jafari, M. Gaba, J. Wiesner, D. G. Wright, Y. Wei, H. Vashisht, Y. Kulizhskaya, J. Hoover, M. Le, L. Li, C. Iwuanyanwu, L. Liu, K. Ramirez, A. Khorlin, A. Cui, T. LIN, M. Wu, R. Aguilar, K. Pallo, A. Chakladar, G. Perng, E. A. Abellan, M. Zhang, I. Dasgupta, N. Kushman, I. Penchev, A. Repina, X. Wu, T. van der Weide, P. Ponnapalli, C. Kaplan, J. Simsa, S. Li, O. Dousse, F. Yang, J. Piper, N. Ie, R. Pasumarthi, N. Lintz, A. Vijayakumar, D. Andor, P. Valenzuela, M. Lui, C. Paduraru, D. Peng, K. Lee, S. Zhang, S. Greene, D. D. Nguyen, P. Kurylowicz, C. Hardin, L. Dixon, L. Janzer, K. Choo, Z. Feng, B. Zhang, A. Singhal, D. Du, D. McKinnon, N. Antropova, T. Bolukbasi, O. Keller, D. Reid, D. Finchelstein, M. A. Raad, R. Crocker, P. Hawkins, R. Dadashi, C. Gaffney, K. Franko, A. Bulanova, R. Leblond, S. Chung, H. Askham, L. C. Cobo, K. Xu, F. Fischer, J. Xu, C. Sorokin, C. Alberti, C.-C. Lin, C. Evans, A. Dimitriev, H. Forbes, D. Banarse, Z. Tung, M. Omernick, C. Bishop, R. Sterneck, R. Jain, J. Xia, E. Amid, F. Piccinno, X. Wang, P. Banzal, D. J. Mankowitz, A. Polozov, V. Krakovna, S. Brown, M. Bateni, D. Duan, V. Firoiu, M. Thotakuri, T. Natan, M. Geist, S. tan Girgin, H. Li, J. Ye, O. Roval, R. Tojo, M. Kwong, J. Lee-Thorp, C. Yew, D. Sinopalnikov, S. Ramos, J. Mellor, A. Sharma, K. Wu, D. Miller, N. Sonnerat, D. Vnukov, R. Greig, J. Beattie, E. Caveness, L. Bai, J. Eisenschlos, A. Korchemniy, T. Tsai, M. Jasarevic, W. Kong, P. Dao, Z. Zheng, F. Liu, F. Yang, R. Zhu, T. H. Teh, J. Sanmiya, E. Gladchenko, N. Trdin, D. Toyama, E. Rosen, S. Tavakkol, L. Xue, C. Elkind, O. Woodman, J. Carpenter, G. Papamakarios, R. Kemp, S. Kafle, T. Grunina, R. Sinha, A. Talbert, D. Wu, D. Owusu-Afriyie, C. Du, C. Thornton, J. Pont-Tuset, P. Narayana, J. Li, S. Fatehi, J. Wieting, O. Ajmeri, B. Uria, Y. Ko, L. Knight, A. Héliou, N. Niu, S. Gu, C. Pang, Y. Li, N. Levine, A. Stolovich, R. Santamaria-Fernandez, S. Goenka, W. Yustalim, R. Strudel, A. Elqursh, C. Deck, H. Lee, Z. Li, K. Levin, R. Hoffmann, D. Holtmann-Rice, O. Bachem, S. Arora, C. Koh, S. H. Yeganeh, S. Põder, M. Tariq, Y. Sun, L. Ionita, M. Seyedhosseini, P. Tafti, Z. Liu, A. Gulati, J. Liu, X. Ye, B. Chrzaszcz, L. Wang, N. Sethi, T. Li, B. Brown, S. Singh, W. Fan, A. Parisi, J. Stanton, V. Koverkathu, C. A. Choquette-Choo, Y. Li, T. Lu, A. Ittycheriah, P. Shroff, M. Varadarajan, S. Bahargam, R. Willoughby, D. Gaddy, G. Desjardins, M. Cornero, B. Robenek, B. Mittal, B. Albrecht, A. Shenoy, F. Moiseev, H. Jacobsson, A. Ghaffarkhah, M. Rivière, A. Walton, C. Crepy, A. Parrish, Z. Zhou, C. Farabet, C. Radebaugh, P. Srinivasan, C. van der Salm,

A. Fidjeland, S. Scellato, E. Latorre-Chimoto, H. Klimczak-Plucińska, D. Bridson, D. de Cesare, T. Hudson, P. Mendolicchio, L. Walker, A. Morris, M. Mauger, A. Guseynov, A. Reid, S. Odoom, L. Loher, V. Cotruta, M. Yenugula, D. Grewe, A. Petrushkina, T. Duerig, A. Sanchez, S. Yadlowsky, A. Shen, A. Globerson, L. Webb, S. Dua, D. Li, S. Bhupatiraju, D. Hurt, H. Qureshi, A. Agarwal, T. Shani, M. Eyal, A. Khare, S. R. Belle, L. Wang, C. Tekur, M. S. Kale, J. Wei, R. Sang, B. Saeta, T. Liechty, Y. Sun, Y. Zhao, S. Lee, P. Nayak, D. Fritz, M. R. Vuyyuru, J. Aslanides, N. Vyas, M. Wicke, X. Ma, E. Eltyshev, N. Martin, H. Cate, J. Manyika, K. Amiri, Y. Kim, X. Xiong, K. Kang, F. Luisier, N. Tripuraneni, D. Madras, M. Guo, A. Waters, O. Wang, J. Ainslie, J. Baldridge, H. Zhang, G. Pruthi, J. Bauer, F. Yang, R. Mansour, J. Gelman, Y. Xu, G. Polovets, J. Liu, H. Cai, W. Chen, X. Sheng, E. Xue, S. Ozair, C. Angermueller, X. Li, A. Sinha, W. Wang, J. Wiesinger, E. Koukoumidis, Y. Tian, A. Iyer, M. Gurumurthy, M. Goldenson, P. Shah, M. Blake, H. Yu, A. Urbanowicz, J. Palomaki, C. Fernando, K. Durden, H. Mehta, N. Momchev, E. Rahimtoroghi, M. Georgaki, A. Raul, S. Ruder, M. Redshaw, J. Lee, D. Zhou, K. Jalan, D. Li, B. Hechtman, P. Schuh, M. Nasr, K. Milan, V. Mikulik, J. Franco, T. Green, N. Nguyen, J. Kelley, A. Mahendru, A. Hu, J. Howland, B. Vargas, J. Hui, K. Bansal, V. Rao, R. Ghiya, E. Wang, K. Ye, J. M. Sarr, M. M. Preston, M. Elish, S. Li, A. Kaku, J. Gupta, I. Pasupat, D.-C. Juan, M. Someswar, T. M., X. Chen, A. Amini, A. Fabrikant, E. Chu, X. Dong, A. Muthal, S. Buthpitiya, S. Jauhari, N. Hua, U. Khandelwal, A. Hitron, J. Ren, L. Rinaldi, S. Drath, A. Dabush, N.-J. Jiang, H. Godhia, U. Sachs, A. Chen, Y. Fan, H. Taitelbaum, H. Noga, Z. Dai, J. Wang, C. Liang, J. Hamer, C.-S. Ferng, C. Elkind, A. Atias, P. Lee, V. Listík, M. Carlen, J. van de Kerkhof, M. Pikus, K. Zaher, P. Müller, S. Zykova, R. Stefanec, V. Gatsko, C. Hirnschall, A. Sethi, X. F. Xu, C. Ahuja, B. Tsai, A. Stefanoiu, B. Feng, K. Dhandhania, M. Katyal, A. Gupta, A. Parulekar, D. Pitta, J. Zhao, V. Bhatia, Y. Bhavnani, O. Alhadlaq, X. Li, P. Danenberg, D. Tu, A. Pine, V. Filippova, A. Ghosh, B. Limonchik, B. Urala, C. K. Lanka, D. Clive, Y. Sun, E. Li, H. Wu, K. Hongtongsak, I. Li, K. Thakkar, K. Omarov, K. Majmundar, M. Alverson, M. Kucharski, M. Patel, M. Jain, M. Zabelin, P. Pelagatti, R. Kohli, S. Kumar, J. Kim, S. Sankar, V. Shah, L. Ramachandruni, X. Zeng, B. Bariach, L. Weidinger, T. Vu, A. Andreev, A. He, K. Hui, S. Kashem, A. Subramanya, S. Hsiao, D. Hassabis, K. Kavukcuoglu, A. Sadovsky, Q. Le, T. Strohman, Y. Wu, S. Petrov, J. Dean, and O. Vinyals. Gemini: A family of highly capable multimodal models, 2025. URL https://arxiv.org/abs/2312.11805.

Gemma Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, L. Rouillard, T. Mesnard, G. Cideron, J. bastien Grill, S. Ramos, E. Yvinec, M. Casbon, E. Pot, I. Penchev, G. Liu, F. Visin, K. Kenealy, L. Beyer, X. Zhai, A. Tsitsulin, R. Busa-Fekete, A. Feng, N. Sachdeva, B. Coleman, Y. Gao, B. Mustafa, I. Barr, E. Parisotto, D. Tian, M. Eyal, C. Cherry, J.-T. Peter, D. Sinopalnikov, S. Bhupatiraju, R. Agarwal, M. Kazemi, D. Malkin, R. Kumar, D. Vilar, I. Brusilovsky, J. Luo, A. Steiner, A. Friesen, A. Sharma, A. Sharma, A. M. Gilady, A. Goedeckemeyer, A. Saade, A. Feng, A. Kolesnikov, A. Bendebury, A. Abdagic, A. Vadi, A. György, A. S. Pinto, A. Das, A. Bapna, A. Miech, A. Yang, A. Paterson, A. Shenoy, A. Chakrabarti, B. Piot, B. Wu, B. Shahriari, B. Petrini, C. Chen, C. L. Lan, C. A. Choquette-Choo, C. Carey, C. Brick, D. Deutsch, D. Eisenbud, D. Cattle, D. Cheng, D. Paparas, D. S. Sreepathihalli, D. Reid, D. Tran, D. Zelle, E. Noland, E. Huizenga, E. Kharitonov, F. Liu, G. Amirkhanyan, G. Cameron, H. Hashemi, H. Klimczak-Plucińska, H. Singh, H. Mehta, H. T. Lehri, H. Hazimeh, I. Ballantyne, I. Szpektor, I. Nardini, J. Pouget-Abadie, J. Chan, J. Stanton, J. Wieting, J. Lai, J. Orbay, J. Fernandez, J. Newlan, J. yeong Ji, J. Singh, K. Black, K. Yu, K. Hui, K. Vodrahalli, K. Greff, L. Qiu, M. Valentine, M. Coelho, M. Ritter, M. Hoffman, M. Watson, M. Chaturvedi, M. Moynihan, M. Ma, N. Babar, N. Noy, N. Byrd, N. Roy, N. Momchev, N. Chauhan, N. Sachdeva, O. Bunyan, P. Botarda, P. Caron, P. K. Rubenstein, P. Culliton, P. Schmid, P. G. Sessa, P. Xu, P. Stanczyk, P. Tafti, R. Shivanna, R. Wu, R. Pan, R. Rokni, R. Willoughby, R. Vallu, R. Mullins, S. Jerome, S. Smoot, S. Girgin, S. Iqbal, S. Reddy, S. Sheth, S. Põder, S. Bhatnagar, S. R. Panyam, S. Eiger, S. Zhang, T. Liu, T. Yacovone, T. Liechty, U. Kalra, U. Evci, V. Misra, V. Roseberry, V. Feinberg, V. Kolesnikov, W. Han, W. Kwon, X. Chen, Y. Chow, Y. Zhu, Z. Wei, Z. Egyed, V. Cotruta, M. Giang, P. Kirk, A. Rao, K. Black, N. Babar, J. Lo, E. Moreira, L. G. Martins, O. Sanseviero, L. Gonzalez, Z. Gleicher, T. Warkentin, V. Mirrokni, E. Senter, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, Y. Matias, D. Sculley, S. Petrov, N. Fiedel, N. Shazeer, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, J.-B. Alayrac, R. Anil, Dmitry, Lepikhin, S. Borgeaud, O. Bachem, A. Joulin, A. Andreev, C. Hardin, R. Dadashi, and L. Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Google. Gemini models, 2025a. URL https://ai.google.dev/gemini-api/docs/models/.

Google. google/gemma-3-4b-it, 2025b. URL https://huggingface.co/google/gemma-3-4b-it.

M. Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022. URL https://arxiv.org/abs/2203.05794.

H. Hammarström, R. Forkel, M. Haspelmath, and S. Bank. Glottolog 5.2, 2025. URL https://doi.org/10.5281/zenodo.15525265.

L. Ji-An, C. Y. Zhou, M. K. Benna, and M. G. Mattar. Linking in-context learning in transformers to human episodic memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=AYDBFxNon4.

M. Kazemi, B. Fatemi, H. Bansal, J. Palowitch, C. Anastasiou, S. V. Mehta, L. K. Jain, V. Aglietti, D. Jindal, P. Chen, N. Dikkala, G. Tyen, X. Liu, U. Shalit, S. Chiappa, K. Olszewska, Y. Tay, V. Q. Tran, Q. V. Le, and O. Firat. Big-bench extra hard, 2025. URL https://arxiv.org/abs/2502.19187.

A. V. Kharkhurin and L. Wei. The role of code-switching in bilingual creativity. *International Journal of Bilingual Education and Bilingualism*, 18(2):153–169, 2015. doi: 10.1080/13670050.2014.884211. URL https://doi.org/10.1080/13670050.2014.884211.

S. Kheder and E. Kaan. Cognitive control in bilinguals: Proficiency and code-switching both matter. *Cognition*, 209:104575, 2021. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2020.104575. URL https://www.sciencedirect.com/science/article/pii/S0010027720303942.

S. Kumar and D. Jurgens. Are rules meant to be broken? understanding multilingual moral reasoning as a computational pipeline with UniMoral. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5890–5912, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.294. URL https://aclanthology.org/2025.acl-long.294/.

S. Lee, S. Kim, M. Seo, Y. Jo, D. Go, H. Hwang, J. Park, X. Yue, S. Welleck, G. Neubig, M. Lee, and M. Seo. The cot encyclopedia: Analyzing, predicting, and controlling how a reasoning model will think, 2025. URL https://arxiv.org/abs/2505.10185.

Y. Li, J. Xin, M. M. Miao, Q. Long, and L. Ungar. The impact of language mixing on bilingual llm reasoning, 2025. URL https://arxiv.org/abs/2507.15849.

Mistral-AI, :, A. Rastogi, A. Q. Jiang, A. Lo, G. Berrada, G. Lample, J. Rute, J. Barmentlo, K. Yadav, K. Khandelwal, K. R. Chandu, L. Blier, L. Saulnier, M. Dinot, M. Darrin, N. Gupta, R. Soletskyi, S. Vaze, T. L. Scao, Y. Wang, A. Yang, A. H. Liu, A. Sablayrolles, A. Héliou, A. Martin, A. Ehrenberg, A. Agarwal, A. Roux, A. Darcet, A. Mensch, B. Bout, B. Rozière, B. D. Monicault, C. Bamford, C. Wallenwein, C. Renaudin, C. Lanfranchi, D. Dabert, D. Mizelle, D. de las Casas, E. Chane-Sane, E. Fugier, E. B. Hanna, G. Delerce, G. Guinet, G. Novikov, G. Martin, H. Jaju, J. Ludziejewski, J.-H. Chabran, J.-M. Delignon, J. Studnia, J. Amar, J. S. Roberts, J. Denize, K. Saxena, K. Jain, L. Zhao, L. Martin, L. Gao, L. R. Lavaud, M. Pellat, M. Guillaumin, M. Felardos, M. Augustin, M. Seznec, N. Raghuraman, O. Duchenne, P. Wang, P. von Platen, P. Saffer, P. Jacob, P. Wambergue, P. Kurylowicz, P. R. Muddireddy, P. Chagniot, P. Stock, P. Agrawal, R. Sauvestre, R. Delacourt, S. Gandhi, S. Subramanian, S. Dalal, S. Gandhi, S. Ghosh, S. Mishra, S. Aithal, S. Antoniak, T. Schueller, T. Lavril, T. Robert, T. Wang, T. Lacroix, V. Nemychnikova, V. Paltz, V. Richard, W.-D. Li, W. Marshall, X. Zhang, and Y. Tang. Magistral, 2025. URL https://arxiv.org/abs/2506.10910.

I. Moshkov, D. Hanley, I. Sorokin, S. Toshniwal, C. Henkel, B. Schifferer, W. Du, and I. Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset, 2025. URL https://arxiv.org/abs/2504.16891.

N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candes, and T. Hashimoto. s1: Simple test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*, 2025. URL https://openreview.net/forum?id=LdH0vrgAHm.

C. Myers-Scotton. *Code-Switching*, chapter 13, pages 217–237. John Wiley & Sons, Ltd, 2017. ISBN 9781405166256. doi: https://doi.org/10.1002/9781405166256.ch13. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405166256.ch13.

L. T. Schächinger Tenés, J. C. Weiner-Bühler, L. Volpin, A. Grob, K. Skoruppa, and R. K. Segerer. Language proficiency predictors of code-switching behavior in dual-language-learning children. *Bilingualism: Language and Cognition*, 26(5):942–958, 2023. doi: 10.1017/S1366728923000081.

S. Singh, A. Romanou, C. Fourrier, D. I. Adelani, J. G. Ngui, D. Vila-Suero, P. Limkonchotiwat, K. Marchisio, W. Q. Leong, Y. Susanto, R. Ng, S. Longpre, W.-Y. Ko, S. Ruder, M. Smith, A. Bosselut, A. Oh, A. F. T. Martins, L. Choshen, D. Ippolito, E. Ferrante, M. Fadaee, B. Ermis, and S. Hooker. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation, 2025. URL https://arxiv.org/abs/2412.03304.

Z. R. Tam. Language matters: How do multilingual input and reasoning paths affect large reasoning models? GitHub repository, June 2025.

Z. R. Tam, C.-K. Wu, Y. Y. Chiu, C.-Y. Lin, Y.-N. Chen, and H. yi Lee. Language matters: How do multilingual input and reasoning paths affect large reasoning models? *arXiv preprint arXiv:2505.17407*, 2025a.

Z. R. Tam, C.-K. Wu, Y. Y. Chiu, C.-Y. Lin, Y.-N. Chen, and H. yi Lee. Language matters: How do multilingual input and reasoning paths affect large reasoning models?, 2025b. URL https://arxiv.org/abs/2505.17407.

A. W. M. Tan, C. Yu, B. L. Long, W. A. Ma, T. Murray, R. D. Silverman, J. D. Yeatman, and M. Frank. Devbench: A multimodal developmental benchmark for language learning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=zogaeVpbaE.

K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, C. Tang, C. Wang, D. Zhang, E. Yuan, E. Lu, F. Tang, F. Sung, G. Wei, G. Lai, H. Guo, H. Zhu, H. Ding, H. Hu, H. Yang, H. Zhang, H. Yao, H. Zhao, H. Lu, H. Li, H. Yu, H. Gao, H. Zheng, H. Yuan, J. Chen, J. Guo, J. Su, J. Wang, J. Zhao, J. Zhang, J. Liu, J. Yan, J. Wu, L. Shi, L. Ye, L. Yu, M. Dong, N. Zhang, N. Ma, Q. Pan, Q. Gong, S. Liu, S. Ma, S. Wei, S. Cao, S. Huang, T. Jiang, W. Gao, W. Xiong, W. He, W. Huang, W. Xu, W. Wu, W. He, X. Wei, X. Jia, X. Wu, X. Xu, X. Zu, X. Zhou, X. Pan, Y. Charles, Y. Li, Y. Hu, Y. Liu, Y. Chen, Y. Wang, Y. Liu, Y. Qin, Y. Liu, Y. Yang, Y. Bao, Y. Du, Y. Wu, Y. Wang, Z. Zhou, Z. Wang, Z. Li, Z. Zhu, Z. Zhang, Z. Wang, Z. Yang, Z. Huang, Z. Huang, Z. Xu, Z. Yang, and Z. Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Q. Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

J. Torregrossa, C. Bongartz, and S. Eisenbeiß. Does 'translanguaging' equal 'reasoning in multiple languages?': Back to the basics of translanguaging as a way forward. *Linguistic Approaches to Bilingualism*, 15(1):92 – 97, 2025. ISSN 18799264. URL http://proxy.lib.umich.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=ufh&AN=182884221&site=ehost-live&scope=site.

Unicode. Languages and scripts, 2025. URL https://www.unicode.org/cldr/charts/47/supplemental/languages_and_scripts.html.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

M. Wang, L. Lange, H. Adel, Y. Ma, J. Strötgen, and H. Schütze. Language mixing in reasoning language models: Patterns, impact, and internal causes, 2025. URL https://arxiv.org/abs/2505.14815.

R. Wang. The dataset distilled from kimi-k1.5. https://huggingface.co/datasets/wangrongsheng/Kimi-K1.5-Distill-data, 2025.

H. Xu, B. Peng, H. Awadalla, D. Chen, Y.-C. Chen, M. Gao, Y. J. Kim, Y. Li, L. Ren, Y. Shen, S. Wang, W. Xu, J. Gao, and W. Chen. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math, 2025. URL https://arxiv.org/abs/2504.21233.

W. Xuan, R. Yang, H. Qi, Q. Zeng, Y. Xiao, A. Feng, D. Liu, Y. Xing, J. Wang, F. Gao, J. Lu, Y. Jiang, H. Li, X. Li, K. Yu, R. Dong, S. Gu, Y. Li, X. Xie, F. Juefei-Xu, F. Khomh, O. Yoshie, Q. Chen, D. Teodoro, N. Liu, R. Goebel, L. Ma, E. Marrese-Taylor, S. Lu, Y. Iwasawa, Y. Matsuo, and I. Li. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation, 2025. URL https://arxiv.org/abs/2503.10497.

A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Y. Ye, Z. Huang, Y. Xiao, E. Chern, S. Xia, and P. Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

Z.-X. Yong, M. F. Adilazuarda, J. Mansurov, R. Zhang, N. Muennighoff, C. Eickhoff, G. I. Winata, J. Kreutzer, S. H. Bach, and A. F. Aji. Crosslingual reasoning through test-time scaling, 2025. URL https://arxiv.org/abs/2505.05408.

A Technical appendices and supplementary material

A.1 Future work

This work can be extended by fine-tuning and open-sourcing a classifier for our taxonomy and extending our approach to develop a general reasoning behavior taxonomy, not limited to code-switching behaviors. Our taxonomy can also be applied to test whether multilingualism and creativity are positively associated in LLMs, as they are in human cognition. More generally, our taxonomy can be applied to measure the multilingual capabilities of different LLMs [Kharkhurin and Wei, 2015, Fernandez-Duque, 2025].

A.2 Limitations and broader impacts

Developed with a bottom-up approach, our taxonomy depends on the models and datasets included in the initial data. While we include diverse languages, domains, tasks, and models, future work should further expand and diversify the dataset coverage.

Our work is also limited by the language proficiency of the human evaluator of the LLM annotations. Our evaluator is a native English speaker with advanced proficiency in Mandarin Chinese, advanced proficiency in German, and elementary proficiency in Spanish. Additionally, they also have a background in linguistics through which they have prior exposure to Arabic, French, Hindi, Italian, Indonesian, Japanese, Korean, Russian, and Swahili. Nevertheless, it's possible that our human evaluation suffers from accuracy losses due to the evaluator's lack of full fluency in all languages evaluated. Future work should rely on native speakers of all languages included in the dataset to evaluate LLM annotations.

Another limitation of this work is that we use a black box approach, looking at only the final model outputs. An alternative direction pursued by Li et al. [2025] would be to apply methods from mechanistic interpretability to understand and steer model code-switching at the level of internal model activations.

This work can improve usability of large language models (LLMs) for multilinguals and speakers of low-resource languages, by yielding actionable insights into how LLMs handle queries in their languages. We do not anticipate any significant negative societal impacts of this work.

A.3 Experiments compute resources

Generating the reasoning examples from the Global MMLU Lite, MMLU ProX Lite, UniMoral, and BBEH datasets used in this work takes approximately 19.3 hours on 4 NVIDIA A100-SXM4-80GB GPUs, 0.4 hours on 2 NVIDIA RTX A5000 GPUs, and 72.8 hours on 10 NVIDIA RTX A6000 GPUs. We also spend approximately $2.40 and 13.3 hours to prompt DeepSeek-R1 through the DeepSeek API[2] and obtain reasoning examples from Global MMLU Lite. All other data used in this work is drawn from previously existing publicly available datasets.

We use $260.55 of Google Cloud credits for accessing Vertex AI, Google Translate, the Gemini API, and Google Cloud Storage. We use these services for the LLM-assisted brainstorming, as part of our topic modeling pipeline, and for LLM-as-judge annotation described in this work, as well as data processing for preliminary and failed experiments. Prompting gemini-2.5-flash for annotation through the Vertex AI Batch Inference service takes approximately 5.4 hours.

Topic modeling using the BERTopic pipeline described in Appendix A.8 (including preliminary experiments used to develop the final pipeline) takes approximately 1 hour on a single NVIDIA RTX A5000 GPU.

All other data pre-/post-processing and analysis is performed using CPUs on servers with the following specifications:

- 160 cores, 3TB memory, Ubuntu 20.04.6 LTS
- 16 cores, 0.5TB memory, Ubuntu 20.04.6 LTS
- 40 cores, 1TB memory, Ubuntu 20.04.6 LTS
- 48 cores, 1TB memory, Ubuntu 22.04.5 LTS
- 64 cores, 867GB memory, Ubuntu 22.04.2 LTS
- 128 cores, 1.4TB memory, Ubuntu 24.04 LTS

A.4 Configuration for generating reasoning examples

We generate reasoning examples on the BBEH, UniMoral, Global MMLU Lite, and MMLU ProX Lite datasets. All other data used in our experiments is drawn from preexisting reasoning example datasets.

[2] `https://api-docs.deepseek.com/`

BBEH Following the recommendations from the model creators,[3] for prompting DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B we set `temperature=0.6, top_p=0.95`. We also set `max_tokens=32768`, which is the maximum allowed by the DeepSeek distilled models.

UniMoral For both Phi-4-mini-reasoning and DeepSeek-R1-Distill Llama-8B, we set `temperature=0.6`, `top_p=0.95`, and `max_tokens=32768`.

Global MMLU Lite For DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-14B, we set `temperature=0.6`, `top_p=0.9`, and `max_tokens=32768`, again following the recommendations from the model creators.[4] For DeepSeek-R1-Distill-Qwen-32B and DeepSeek-R1-Distill-Llama-70B, we set `temperature=1.0`, `top_p=1.0`, and `max_tokens=32768`. For prompting DeepSeek-R1, we leave the API defaults as is and use the January 2025 model checkpoint.[5]

MMLU ProX Lite For Llama-3.1-Nemotron-Nano-8B-v1, we set `temperature=0.6`, `top_p=0.95`, and `max_tokens=32768`, following NVIDIA's recommendations.[6] We also turn reasoning mode on. For gemma-3-4b-it and DeepSeek-R1-Distill-Qwen-7B, we set `temperature=0` and `max_tokens=2048`, as we initially followed the defaults in `lm-evaluation-harness`.[7] For DeepSeek-R1-Distill-Llama-8B, we set `temperature=0` and `max_tokens=32768`.

A.5 Dataset

Table 1 shows the sources of the data used to develop the taxonomy of code-switched reasoning behaviors. Table 2 shows the sources of the data used to characterize reasoning behaviors.

Table 1: Data used to develop the taxonomy of code-switched reasoning behaviors.

Prompt language	Model	Dataset	Count
ar	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ar	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
ar	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
ar	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
ar	Phi-4-mini-reasoning	UniMoral	50
ar	gemma-3-4b-it	MMLU ProX Lite	50
bn	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
bn	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
bn	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	7
bn	gemma-3-4b-it	MMLU ProX Lite	50
de	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
de	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
de	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
de	gemma-3-4b-it	MMLU ProX Lite	50
en	Claude 3.7 Sonnet	s1K-claude-3-7-sonnet	50
en	DeepSeek-R1	Global MMLU Lite	50

[3] `https://huggingface.co/deepseek-ai/DeepSeek-R1`

[4] `https://huggingface.co/deepseek-ai/DeepSeek-R1`

[5] `https://api-docs.deepseek.com/news/news250120`

[6] `https://build.nvidia.com/nvidia/llama-3_1-nemotron-nano-8b-v1/modelcard`

[7] `https://github.com/EleutherAI/lm-evaluation-harness/blob/main/lm_eval/tasks/mmlu_prox/en/_en_template_yaml`

Prompt language	Model	Dataset	Count
en	DeepSeek-R1	OpenMathReasoning	50
en	DeepSeek-R1	s1K-1.1	50
en	DeepSeek-R1-Distill-Llama-70B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Llama-8B	BBEH	50
en	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
en	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
en	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-32B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-7B	BBEH	50
en	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
en	Kimi k1.5	Kimi-K1.5-Distill-data	50
en	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
en	Phi-4-mini-reasoning	UniMoral	50
en	QwQ-32B	OpenMathReasoning	50
en	gemini-2.0-flash-thinking-exp-1219	s1K-1.1	50
en	gemma-3-4b-it	MMLU ProX Lite	50
en	magistral-small-2506	s1k-magistral-small-2506	50
es	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
es	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
es	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
es	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
es	Phi-4-mini-reasoning	UniMoral	50
es	gemma-3-4b-it	MMLU ProX Lite	50
fr	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
fr	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
fr	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
fr	gemma-3-4b-it	MMLU ProX Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
hi	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
hi	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	5
hi	Phi-4-mini-reasoning	UniMoral	50
hi	gemma-3-4b-it	MMLU ProX Lite	50
id	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ja	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
ja	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50

Prompt language	Model	Dataset	Count
ja	gemma-3-4b-it	MMLU ProX Lite	50
ko	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ko	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
ko	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
ko	gemma-3-4b-it	MMLU ProX Lite	50
my	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
pt	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
pt	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
pt	gemma-3-4b-it	MMLU ProX Lite	50
ru	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
ru	Phi-4-mini-reasoning	UniMoral	50
sw	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
sw	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
sw	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	7
sw	gemma-3-4b-it	MMLU ProX Lite	50
th	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
th	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
th	gemma-3-4b-it	MMLU ProX Lite	50
yo	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
zh	DeepSeek-R1	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-70B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
zh	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-32B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
zh	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
zh	Phi-4-mini-reasoning	UniMoral	50
zh	gemma-3-4b-it	MMLU ProX Lite	50

Prompt language	Model	Dataset	Count
ar	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ar	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
ar	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ar	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50

Prompt language	Model	Dataset	Count
ar	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	20
ar	Phi-4-mini-reasoning	UniMoral	50
ar	gemma-3-4b-it	MMLU ProX Lite	50
bn	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
bn	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
bn	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
bn	gemma-3-4b-it	MMLU ProX Lite	50
de	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
de	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
de	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
de	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
de	gemma-3-4b-it	MMLU ProX Lite	50
en	Claude 3.7 Sonnet	s1K-claude-3-7-sonnet	50
en	DeepSeek-R1	Global MMLU Lite	50
en	DeepSeek-R1	OpenMathReasoning	50
en	DeepSeek-R1	s1K-1.1	50
en	DeepSeek-R1-Distill-Llama-70B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Llama-8B	BBEH	50
en	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
en	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
en	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-32B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-7B	BBEH	50
en	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
en	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
en	Kimi k1.5	Kimi-K1.5-Distill-data	50
en	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
en	Phi-4-mini-reasoning	UniMoral	50
en	QwQ-32B	OpenMathReasoning	50
en	gemini-2.0-flash-thinking-exp-1219	s1K-1.1	50
en	gemma-3-4b-it	MMLU ProX Lite	50
en	magistral-small-2506	s1k-magistral-small-2506	50
es	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
es	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
es	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
es	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
es	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
es	Phi-4-mini-reasoning	UniMoral	50
es	gemma-3-4b-it	MMLU ProX Lite	50
fr	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
fr	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
fr	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
fr	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
fr	gemma-3-4b-it	MMLU ProX Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
hi	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
hi	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50

Prompt language	Model	Dataset	Count
hi	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
hi	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
hi	Phi-4-mini-reasoning	UniMoral	50
hi	gemma-3-4b-it	MMLU ProX Lite	50
id	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
id	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
it	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ja	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ja	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
ja	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
ja	gemma-3-4b-it	MMLU ProX Lite	50
ko	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
ko	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
ko	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
ko	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	28
ko	gemma-3-4b-it	MMLU ProX Lite	50
my	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
my	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
pt	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
pt	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
pt	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	44
pt	gemma-3-4b-it	MMLU ProX Lite	50
ru	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
ru	Phi-4-mini-reasoning	UniMoral	50
sw	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
sw	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
sw	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
sw	gemma-3-4b-it	MMLU ProX Lite	50
th	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
th	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
th	gemma-3-4b-it	MMLU ProX Lite	50
yo	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50
yo	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
zh	DeepSeek-R1	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-70B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	MMLU ProX Lite	50
zh	DeepSeek-R1-Distill-Llama-8B	UniMoral	50
zh	DeepSeek-R1-Distill-Qwen-1.5B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-14B	Global MMLU Lite	50

Prompt language	Model	Dataset	Count
zh	DeepSeek-R1-Distill-Qwen-32B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-7B	Global MMLU Lite	50
zh	DeepSeek-R1-Distill-Qwen-7B	MMLU ProX Lite	50
zh	Llama-3.1-Nemotron-Nano-8B-v1	MMLU ProX Lite	50
zh	Phi-4-mini-reasoning	UniMoral	50
zh	gemma-3-4b-it	MMLU ProX Lite	50

Table 2: Data used to characterize model behaviors.

A.6 Full taxonomy of code-switched reasoning behaviors

The taxonomy is shown in Table 3.

A.7 Prompt for brainstorming code-switched reasoning classification criteria

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

You are tasked with analyzing any code-switching strategies used
    in the above reasoning, where code-switching is defined as the
     use of multiple scripts or languages. Your goal is to extract
     and describe patterns based on various criteria that
    characterize how the model uses code-switching in its
    reasoning.

Please follow these guidelines:

1. Identify multiple *meaningful criteria* that differentiate code
    -switching strategies. Each criterion should have a clear and
    descriptive name that reflects a real aspect of the code-
    switching process. **Do not use generic placeholders like '
    Criterion 1'.**

2. For each criterion, provide a brief description.

3. Present your analysis in the following format, using <patterns>
     and </patterns> tags to enclose the list:

<patterns>
Descriptive Criterion Name (Description of the Criterion)
Descriptive Criterion Name (Description of the Criterion)
...
Descriptive Criterion Name (Description of the Criterion)
</patterns>

4. Do not include any additional explanations or commentary within
     the <patterns> tags.

5. If no code-switching is observed in the reasoning, simply
    return "<patterns></patterns>".
```

Table 3: Annotation schema for analyzing multilingual reasoning behaviors.

Dimension	**Name:** Description	**Granularity**
Function	**Translation and Interpretation**: Translates or interprets foreign language phrases into the primary language of reasoning to understand meaning.	Step-level
	Lexical Ambiguity Resolution: Identifies unclear terms in one language and clarifies meaning in another.	Step-level
	Parenthetical Translation: Provides a direct translation of a single word or short phrase in parentheses, often immediately after the term appears, reinforcing the precise meaning of a concept for clarity and accuracy.	Step-level
	Lexical Retention: Retains words/phrases from the original language of the prompt, without translating them to the primary language of reasoning.	Step-level
Form	**Directionality of Code-switching**: Whether the code-switching is from the primary language of reasoning to a second language, or vice-versa.	Switch-level
	Scope of Code-switching: Whether the code-switching occurs at the level of individual words, phrases, sentences, or entire reasoning steps.	Word-/Phrase-/Sentence-/Step-/Instance-level
	Language-Role Mapping: Which language is used for different aspects of the reasoning process.	Instance-level
	Language of Core Reasoning: The primary language used for the logical progression and problem-solving steps of the reasoning process.	Instance-level
	Language of Internal Monologue: The language in which the model uses conversational phrases and self-correction markers common in human thought processes, such as "Okay, so...", "Let's see.", "Wait,", "Hmm.", "No.", "Alright," to structure its reasoning and signal its thought progression.	Instance-level
	Language of Problem Deconstruction: The language used when breaking down or rephrasing parts of the original problem statement or options.	Instance-level
	Language of Self-Correction: The language used by the model during self-correction or re-evaluation of its understanding.	Instance-level
	Language of External Knowledge Retrieval: The language used when recalling or integrating external information or general knowledge relevant to the problem.	Instance-level
	Language-Content Mapping: Maps how different languages are used for different content types.	Step-level
	Language of Direct Quotes: Language used when quoting original problem text.	Step-level
	Language of Technical Terminology: Language for domain-specific terms and concepts.	Step-level
	Language of Final Answer: Language used to state the final answer or conclusion.	Step-level
	Interleaving Natural and Non-natural Language Elements: Mixing elements such as code and math into natural language.	Step-level
	Interleaving Math and Natural Language: Integration of mathematical symbols, equations, or expressions directly within natural language sentences or phrases.	Step-level
	Interleaving Code and Natural Language: Integration of elements such as code snippets, programming language constructs, or pseudocode directly within natural language sentences or phrases.	Step-level
Coherence	**Fluency of Code-Switching**: Assesses naturalness and accuracy of language switching.	Instance-level
	Consistency of Terminology: Whether specific terms, once introduced in one language, are consistently maintained in that language or switch back and forth.	Instance-level

A.8 BERTopic modeling details

All components of our topic modeling pipeline used in the development of the code-switched reasoning taxonomy are implemented in BERTopic. We use all-mpnet-base-v2 to embed the text of each criterion, UMAP for dimension reduction, HDBSCAN for clustering, and Gemini 2.5 Flash for generating topic representations. For UMAP dimension reduction, we set `n_neighbors=15, n_components=5, min_dist=0.0, metric='cosine'`. For HDBSCAN clustering, we set `min_cluster_size=15, metric='euclidean', cluster_selection_method='eom'`, and `prediction_data=True`. We use `CountVectorizer` with `stop_words=''english''` for topic tokenization and c-TF-IDF for extracting topic words. Finally, we Gemini 2.5 Flash to generate more interpretable topic representations.

A.9 Prompt for annotating the languages used for core reasoning, internal monologue, problem deconstruction, self-correction, and external knowledge retrieval

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

You are tasked with identifying the languages used for different
    roles in the reasoning process.

The roles are defined as follows, in the format "<role>: <
    role_definition>"
```
{{
    "Language of Core Reasoning": "The primary language used for
        the logical progression and problem-solving steps of the
        reasoning process.",
    "Language of Internal Monologue": "The language in which the
        model uses conversational phrases and self-correction
        markers common in human thought processes, such as \"Okay,
         so...\", \"Let's see.\", \"Wait,\", \"Hmm.\", \"No.\", \"
        Alright,\" to structure its reasoning and signal its
        thought progression.",
    "Language of Problem Deconstruction": "The language used when
        breaking down or rephrasing parts of the original problem
        statement or options.",
    "Language of Self-Correction": "The language used by the model
         during self-correction or re-evaluation of its
        understanding.",
    "Language of External Knowledge Retrieval": "The language used
         when recalling or integrating external information or
        general knowledge relevant to the problem."
}}
```

The languages are defined as follows, in the format "<
    language_name>: <language_code>":
```
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
```
```

````
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
```

Identify the primary language used for each role in the reasoning process, outputting your answer using the above two-letter ISO 639 language codes.
If you identify a language not listed above, please use the appropriate two-letter ISO 639 language code for that language.
If you are unable to identify a language for a role, please return "None" for that role.

Return your answer in the following JSON format:
```
{{
    "core_reasoning_language": "<language_code | None>",
    "internal_monologue_language": "<language_code | None>",
    "problem_deconstruction_language": "<language_code | None>",
    "self_correction_language": "<language_code | None>",
    "external_knowledge_retrieval_language": "<language_code | None>"
}}
```
````

A.10 Prompt for annotating lexical ambiguity resolution

````
Here is a problem and the reasoning process that a model generated when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning process has been split into distinct reasoning steps in the format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, you are tasked with determining whether the model performs Lexical Ambiguity Resolution.

Lexical Ambiguity Resolution occurs when the model identifies terms that are unclear or ambiguous in one language and attempts to clarify their meaning in another language.

For each step in which Lexical Ambiguity Resolution occurs, you should also determine which language the unclear/ambiguous
````
````

```
    terms are originally in ("ambiguous_language"), and which
    language is used for the Lexical Ambiguity Resolution ("
    resolution_language").

The languages are defined as follows, in the format "<
    language_name>: <language_code>":
```
```
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
```
```
If you identify a language not listed above, please use the
    appropriate two-letter ISO 639 language code for that language
    .
If you are unable to identify a language for a role, please return
     "None" for that role.

Return your answer in the following JSON format:
```
```
{{
    [<step_idx>: {{"ambiguous_language": "<language_code | None>",
                   "resolution_language": "<language_code | None>"
                      }},
    <step_idx>: {{"ambiguous_language": "<language_code | None>",
                   "resolution_language": "<language_code | None>"
                      }},
                   ...]
}}
```
```
```

A.11 Prompt for annotating parenthetical translation

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
```
{reasoning}
```
```
```

```
For each step, you are tasked with identifying any cases of
    Parenthetical Translation.

Parenthetical Translation occurs when the model provides a direct
    translation of a single word or short phrase in parentheses,
    often immediately after the term appears, reinforcing the
    precise meaning of a concept for clarity and accuracy.

For each step in which Parenthetical Translation occurs, you
    should extract the original term ("<original_term>"), its
    parenthetical translation ("<parenthetical_translation>"), the
     source language of the original term ("<source_language>"),
    and the target language of the parenthetical translation ("<
    target_language>").

Use the following language codes, provided in the format "<
    language_name>: <language_code>":
'''
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
'''

If you identify a language not listed above, please use the
    appropriate two-letter ISO 639 language code for that language
    .
If you are unable to identify a language for a role, please return
     "None" for that role.

Return your answer in the following JSON format:
'''
{{
    [<step_idx>: [{{"original_term": "<original_term>",
                    "parenthetical_translation": "<
                       parenthetical_translation>",
                    "source_language": "<language_code | None>",
                    "target_language": "<language_code | None>"}},
                  {{"original_term": "<original_term>",
                    "parenthetical_translation": "<
                       parenthetical_translation>",
                    "source_language": "<language_code | None>",
                    "target_language": "<language_code | None>"}},
                    ...],
    <step_idx>: [{{"original_term": "<original_term>",
                    "parenthetical_translation": "<
                       parenthetical_translation>",
                    "source_language": "<language_code | None>",
```

```
                    "target_language": "<language_code | None>"}},
                  {{"original_term": "<original_term>",
                    "parenthetical_translation": "<
                        parenthetical_translation>",
                    "source_language": "<language_code | None>",
                    "target_language": "<language_code | None>"}},
                    ...],
}}
```

A.12 Prompt for annotating lexical retention

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, you are tasked with identifying any cases of
    Lexical Retention.

Lexical Retention occurs when the model retains specific words or
    phrases from the original language of the problem within its
    reasoning, without translating them to the primary language of
     reasoning.

For each step in which Lexical Retention occurs, you should
    extract any retained terms ("<retained_term>") and the
    translation of each retained term to English ("<
    english_translation>").

Return your answer in the following JSON format:
```
{{
    [<step_idx>: [{{"retained_term": "<retained_term>",
                    "english_translation": "<english_translation>"
                        }},
                  {{"retained_term": "<retained_term>",
                    "english_translation": "<english_translation>"
                        }},
                    ...],
    <step_idx>: [{{"retained_term": "<retained_term>",
                    "english_translation": "<english_translation>"
                        }},
                  {{"retained_term": "<retained_term>",
                    "english_translation": "<english_translation>"
                        }},
                    ...],
}}
```
```

A.13 Prompt for annotating direct quotes

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, you are tasked with identifying any cases of Direct
     Quotation.

Direct Quotation occurs when the model repeats parts of the
    original problem statement in its reasoning word-for-word.

The quotes can be in the original language of the problem or in
    another language, and they may be enclosed in quotation marks
    or not.

For each step in which Direct Quotation occurs, you should extract
     each quoted span of text ("<quotation_text>") and the
    language of the quoted text ("<quotation_language>").

Use the following language codes, provided in the format "<
    language_name>: <language_code>":
```
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
```

If you identify a language not listed above, please use the
    appropriate two-letter ISO 639 language code for that language
    .
If you are unable to identify the language, please return "None."

Return your answer in the following JSON format:
```
{{
    [<step_idx>: [{{"quotation_text": "<quotation_text>",
                    "quotation_language": "<language_code | None>"
                        }},
```
```

```
                    {{"quotation_text": "<quotation_text>",
                      "quotation_language": "<language_code | None>"
                          }},
                      ...],
    <step_idx>: [{{"quotation_text": "<quotation_text>",
                      "quotation_language": "<language_code | None>"
                          }},
                    {{"quotation_text": "<quotation_text>",
                      "quotation_language": "<language_code | None>"
                          }},
                      ...],
}}
```

A.14 Prompt for annotating language of technical terms

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, identify any terms used by the model that are
    specialized or specific to the problem domain ("term").

For each identified term, identify the language of the term ("
    language").

Use the following language codes, provided in the format "<
    language_name>: <language_code>":
```
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
```
```

```
If you identify a language not listed above, please use the
    appropriate two-letter ISO 639 language code for that language
    .
If you are unable to identify the language, please return "None."

Return your answer in the following JSON format:
```
{{
    [<step_idx>: [{{"term": "<term>",
                    "language": "<language_code | None>"}},
                  {{"term": "<term>",
                    "language": "<language_code | None>"}},
                    ...],
    <step_idx>: [{{"term": "<term>",
                    "language": "<language_code | None>"}},
                  {{"term": "<term>",
                    "language": "<language_code | None>"}},
                    ...],
}}
```
```

A.15 Prompt for annotating language of final answer

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, first determine if the model provides a final
    answer to the problem. Keep in mind that the model may reach a
     final answer at multiple points in the reasoning process and
    may even change its answer throughout.

Second, each time the model provides a final answer, identify the
    language used to state the final answer.

Use the following language codes, provided in the format "<
    language_name>: <language_code>":
```
{{
    "Arabic": "ar",
    "Bengali": "bn",
    "German": "de",
    "English": "en",
    "Spanish": "es",
    "French": "fr",
    "Hindi": "hi",
    "Indonesian": "id",
    "Italian": "it",
    "Japanese": "ja",
    "Korean": "ko",
    "Burmese": "my",
    "Portuguese": "pt",
    "Russian": "ru",
```
```

```
    "Swahili": "sw",
    "Thai": "th",
    "Yoruba": "yo",
    "Chinese": "zh",
}}
```

```
If you identify a language not listed above, please use the
    appropriate two-letter ISO 639 language code for that language
    .
If you are unable to identify the language, please return "None."

Return your answer in the following JSON format:
```
```
{{<step_idx>: "<language_code | None>",
  <step_idx>: "<language_code | None>",
   ...
}}
```
```

### A.16 Prompt for annotating code-switching between math and natural language

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
```
{reasoning}
```
```

For each step, determine if the model integrates mathematical
    symbols, equations, or expressions directly within natural
    language sentences or phrases.

If the model does integrate math with natural language, answer "
    True" for that step. Otherwise, answer "False".

Return your answer in the following JSON format:
```
```
{{<step_idx>: <True | False>,
  <step_idx>: <True | False>,
   ...
}}
```
```

A.17 Prompt for annotating code-switching between programming and natural languages

```
Here is a problem and the reasoning process that a model generated
     when it tried to solve the problem.

Problem: (enclosed in double backticks)
``
{problem}
``
```

````
Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

For each step, determine if the model integrates code with natural
     language.

This could look like embedding code snippets, programming language
     constructs, or pseudocode directly within natural language
    sentences or phrases.

If the model does integrate code with natural language, answer "
    True" for that step. Otherwise, answer "False".

Return your answer in the following JSON format:
```
{{<step_idx>: <True | False>,
  <step_idx>: <True | False>,
   ...
}}
```
````

A.18 Prompt for annotating presence of word-, phrase-, sentence-, step-, and instance-level code-switching

````
Here is the reasoning process that a model generated when it tried
     to solve a problem.

Problem: (enclosed in double backticks)
``
{problem}
``

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

Determine if the model performs any of the following types of code
    -switching in its reasoning process and respond with either "
    True" or "False" for each type:
- word_level: The model inserts individual words from one language
    /script into a sentence primarily in another language/script.
- phrase_level: Phrases within a single sentence are in different
    languages/scripts.
- sentence_level: Within a given reasoning step, entire sentences
    are in different languages/scripts, but the sentences are not
    mixed within the same sentence.
- step_level: Out of all reasoning steps, there are at least two
    reasoning steps where the model uses different languages/
    scripts, but the steps themselves are not mixed within the
    same step.
- instance_level: The model uses only a single language/script
    throughout the entire reasoning process, but this language/
    script is different from the language/script of the problem
    statement.

Return your answer in the following JSON format:
```
````
````

```
{{"word_level": <True | False>,
  "phrase_level": <True | False>,
  "sentence_level": <True | False>,
  "step_level": <True | False>,
  "instance_level": <True | False>
}}
```
```

### A.19 Prompt for annotating code-switching fluency

```
Here is the reasoning process that a model generated when it tried
     to solve a problem.

Reasoning process: (enclosed in triple backticks, the reasoning
    process has been split into distinct reasoning steps in the
    format of <step_idx><reasoning_step_content></step_idx>)
```
{reasoning}
```

First, determine if the reasoning contains any code-switching,
    defined as the use of multiple scripts or languages within the
     reasoning process.
Then, rate the overall fluency of the code-switching in the
    reasoning process on a scale from 1 to 5, where fluency
    describes the accuracy and naturalness of the code-switching.
The scale is defined as follows, in the format <index>. <
    description>:

0. No code-switching
1. Very disfluent
2. Somewhat disfluent
3. Neither fluent nor disfluent
4. Somewhat fluent
5. Very fluent

Return your answer as "<index>".
```

### A.20 Human evaluation of LLM annotations

For validating the LLM-brainstormed criteria to differentiate code-switching strategies, we follow Lee et al. [2025] in asking our evaluator to answer the following yes-no question for each instance: "Are the automatically generated, detailed criteria plausible?"

For evaluating the quality of our LLM annotations of fluency (a continuous measure), we compute the Pearson correlation coefficient between human and LLM ratings with `scipy` [Virtanen et al., 2020]. For hypothesis testing, we assume that the correlation is positive as our alternative hypothesis. We include those instances labeled with "0" for the fluency score (indicating no code-switching detected). We exclude those instances with malformed response formats that prevent accurate parsing of the score. The rate of malformed responses for fluency annotations is $1\%$.

For the binary labels in Table 4, the rate of malformed responses is $0\%$.

Table 4: Evaluation metrics for LLM annotations of code-switching behaviors highlighted in this work. Here, we evaluate binary classification, where each reasoning instance is labeled as either featuring (1) or not featuring (0) each of the listed behaviors. For each annotated behavior, the naive baseline is to guess the majority label (0 or 1).

| Behavior | Precision | Recall | Micro F1/Accuracy | Macro F1 |
|---|---|---|---|---|
| Parenthetical translation (LLM) | **0.1500** | **1.0000** | 0.8300 | **0.5824** |
| Parenthetical translation (Baseline) | 0.0000 | 1.0000 | **0.9700** | 0.4924 |
| Lexical ambiguity resolution (LLM) | **0.5000** | **0.5000** | **0.8200** | **0.6951** |
```

Behavior	Precision	Recall	Micro F1/Accuracy	Macro F1
Lexical ambiguity resolution (Baseline)	0.0000	0.0000	0.8200	0.4505
Lexical retention (LLM)	**0.1594**	**0.9167**	0.4100	0.3879
Lexical retention (Baseline)	0.0000	0.0000	**0.8800**	**0.4681**
Instance-level code-switching (LLM)	**0.5417**	**0.8125**	**0.8600**	**0.7812**
Instance-level code-switching (Baseline)	0.0000	0.0000	0.8400	0.4565
Step-level code-switching (LLM)	**0.0857**	**1.0000**	0.6800	0.4802
Step-level code-switching (Baseline)	0.0000	0.0000	**0.9700**	**0.4924**
Sentence-level code-switching (LLM)	**0.0000**	**0.0000**	0.9500	0.4872
Sentence-level code-switching (Baseline)	0.0000	0.0000	**0.9900**	**0.4975**
Phrase-level code-switching (LLM)	**0.6111**	**0.4400**	**0.7900**	**0.6889**
Phrase-level code-switching (Baseline)	0.0000	0.0000	0.7500	0.4286
Word-level code-switching (LLM)	**0.5385**	**0.5385**	**0.7600**	**0.6881**
Word-level code-switching (Baseline)	0.0000	0.0000	0.7400	0.4253
Average (LLM)	**0.3233**	**0.6510**	0.7625	**0.5989**
Average (Baseline)	0.0000	0.1250	**0.8700**	0.4639

A.21 Languages supported by models

Table 5: Languages supported and unsupported by models in our dataset

Model	Supported languages	Unsupported languages
DeepSeek-R1-Distill-Qwen-1.5B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1-Distill-Qwen-7B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1-Distill-Llama-8B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1-Distill-Qwen-14B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1-Distill-Qwen-32B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1-Distill-Llama-70B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
DeepSeek-R1	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
Llama-3.1-Nemotron-Nano-8B-v1	en, de, fr, it, pt, es, th, hi	zh, ar, ja, ko, bn, sw, id, yo, my, ru
phi-4-mini-reasoning	en	zh, es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru
gemma-3-4b-it	140+ languages[8]	Unknown
magistral-small-2506	en, fr, de, el, hi, id, it, ja, ko, ms, ne, pl, pt, ro, ru, sr, es, tr, uk, vi, ar, bn, zh, fa	sw, yo, my, th
gemini-2.0-flash-thinking-exp-1219	ar, bn, bg, zh, hr, cs, da, nl, en, et, fi, fr, de, el, iw, hi, hu, id, it, ja, ko, lv, lt, no, pl, pt, ro, ru, sr, sk, sl, es, sw, sv, th, tr, uk, vi	None
Claude 3.7 Sonnet	en, es, pt, it, fr, id, de, ar, zh, ko, ja, hi, bn, sw, yo	None
QwQ-32B	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru

[8]The Gemma 3 paper and website do not specify which languages are supported, but we assume that the over 140 languages covered by the model include those languages that we prompt the model on in this work [Google, 2025b, Gemma Team et al., 2025].

Model	Supported languages	Unsupported languages
Kimi 1.5	en, zh	es, ar, hi, de, fr, pt, ja, ko, bn, sw, it, id, yo, my, th, ru

A.22 Details of generalized linear mixed effects analysis

For the full implementation of our generalized linear mixed effects analysis, see our code at `https://figshare.com/s/f8f6fd2d899b93077b03`. We use `glmer` from the `lme4` R library for modeling, setting `family = binomial(link = ''logit'')`.[9] We use R's base implementation of ANOVA from the `stats` package to perform significance testing.[10]

For the model presented in Figure 3, the response variable is the presence/absence of each code-switched reasoning behavior listed in Table 6, coded as a binary variable. Whether the prompt language is officially supported by the model is coded as a binary variable and included as a fixed effect. The dataset item identifier and model (both categorical variables) are included as random effects. We assume random slopes and intercepts for the random effects. For significance testing, for each response variable we construct a null model without the fixed effect of prompt language.

Table 6: ANOVA results for generalized linear mixed effects models of effect of prompt language.

Response variable	χ^2	df	p	Increase in log odds	SE
Parenthetical translation	14.68	1	0.0001274	2.4489	0.5422
Lexical ambiguity resolution	25.985	1	3.441×10^{-7}	3.4704	0.5228
Lexical retention	16.395	1	5.142×10^{-5}	1.8906	0.3095
Word-level code-switch	19.955	1	7.93×10^{-6}	3.1296	0.5473
Phrase-level code-switch	21.552	1	3.443×10^{-6}	2.9362859	0.0005249
Sentence-level code-switch	26.632	1	2.461×10^{-7}	4.331	1.014
Step-level code-switch	26.304	1	2.916×10^{-7}	3.9072	0.5499
Instance-level code-switch	25.93	1	3.54×10^{-7}	3.3258	0.5182

For modeling the effect of code-switching fluency on accuracy, we treat the correctness of the final answer as the binary response variable. We include an interaction term for the joint effect of fluency (rated on a Likert scale from one to five) and whether or not the prompt language is supported by the model (binary variable). The dataset item and model (both categorical variables) are treated as random effects. For significance testing, we construct a null model omitting the effect of fluency. We only include those instances that feature code-switching in their reasoning for modeling ($n = 2054$). Each one-point increase in the fluency rating of a model's code-switching during reasoning increases the log odds of a correct final answer by 0.11 ± 0.05 (standard errors), ($\chi^2(2) = 12.16, p = 0.002292$).

A.23 Significance testing for difference in fluency scores between correct and incorrect instances

We perform a one-tailed Welch's t-test[11] and report the results in Table 7.

Table 7: Results of significance testing for difference in fluency scores between correct and incorrect instances.

Comparison	M_1	SD_1	M_2	SD_2	t	df	p
Incorrect v. correct instances (unsupported prompt language)	3.00	1.32	3.88	1.17	4.04	41.97	0.00011
Incorrect v. correct instances (supported prompt language)	2.61	1.20	2.86	1.16	3.94	860.9	4.48×10^{-5}
Incorrect v. correct instances (all prompt languages)	2.68	1.24	2.93	1.19	3.98	874.33	3.80×10^{-5}

[9] `https://search.r-project.org/CRAN/refmans/lme4/html/glmer.html`

[10] `https://stat.ethz.ch/R-manual/R-devel/library/stats/html/anova.html`

[11] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html`

A.24 Sources of code used in this work

Parts of our code (available at `https://figshare.com/s/f8f6fd2d899b93077b03`) are sourced from the following works: Tam et al. [2025a], Tam [2025], Gao et al. [2024], Grootendorst [2022], and Kumar and Jurgens [2025].

A.25 Models used to source reasoning examples

Table 8: Sources of models used to generate reasoning examples in this work.

Model	Citation
DeepSeek-R1 and its distilled variants	DeepSeek-AI et al. [2025]
gemini-2.0-flash-thinking-exp-1219	Google [2025a], Gemini Team et al. [2025]
Claude Sonnet 3.7	Anthropic [2025]
Llama-3.1-Nemotron-Nano-8B-v1	Bercovich et al. [2025]
gemma-3-4b-it	Google [2025b]
Phi-4-mini-reasoning	Xu et al. [2025]
QwQ-32B	Yang et al. [2024], Team [2025]
Magistral Small 1.0	Mistral-AI et al. [2025]
Kimi k1.5	Team et al. [2025]

A.26 Datasets used to source reasoning examples

Table 9: Datasets used to generate reasoning examples in this work.

Dataset	Type	Citation
s1K-1.1 and s1K-claude-3-7-sonnet	Coding, math, puzzle, science, legal, logic, humanities	Muennighoff et al. [2025]
Global MMLU	Math, science, humanities, medical, business, legal	Singh et al. [2025]
MMLU ProX	Math, science, humanities, medical, business, legal	Xuan et al. [2025]
BIG-Bench Extra Hard	Puzzle, logic, commonsense, linguistics, spatial	Kazemi et al. [2025]
UniMoral	Cultural, social	Kumar and Jurgens [2025]
OpenMathReasoning	Math	Moshkov et al. [2025]
Kimi-K1.5-Distill-data	Math	Wang [2025], Ye et al. [2025]

A.27 Additional examples of alignment between model and human reasoning behaviors

Figure 4 includes an additional example of alignment between human and model code-switched reasoning behaviors.

A.28 Additional visualizations of effect of prompting in unsupported languages

In Figure 5, we compare frequencies of diverse code-switching behaviors for instances where a model is queried in a language it officially supports, versus in an unsupported language. (Error bars represent binomial standard error.)

As seen in Figure 6, in cases where a model uses code-switching in its reasoning, higher code-switching fluency is associated with higher accuracy (see Appendix A.23 for details of significance testing). Instances with correct final answers feature reasoning that is rated as significantly more fluent on a scale from 1 to 5 than instances with incorrect answers ($p < 0.001$). Looking separately at instances where the model is prompted in supported and unsupported languages, we find that the difference in fluency between successful and unsuccessful reasoning is even more pronounced when the model is handling queries in unsupported languages ($p < 0.001$). In contrast, the difference in mean fluency for correct and incorrect instances when the model is prompted in supported languages is smaller, though still significant, ($p < 0.001$). The positive association between code-switching fluency in model reasoning and accuracy suggests that code-switching that is more natural and human-like may be more beneficial to model performance.

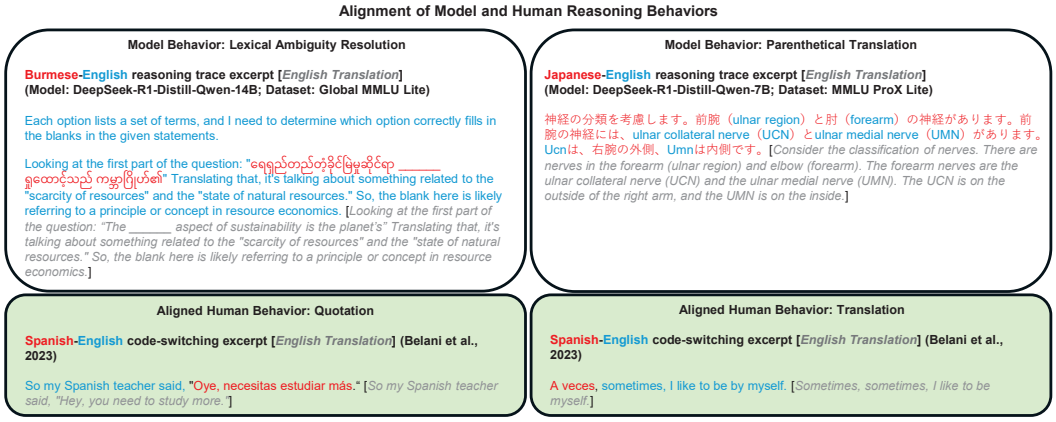

Figure 4: Examples of parallels between human code-switching and code-switching in model reasoning. Human code-switching examples are from [Belani and Flanigan, 2023].

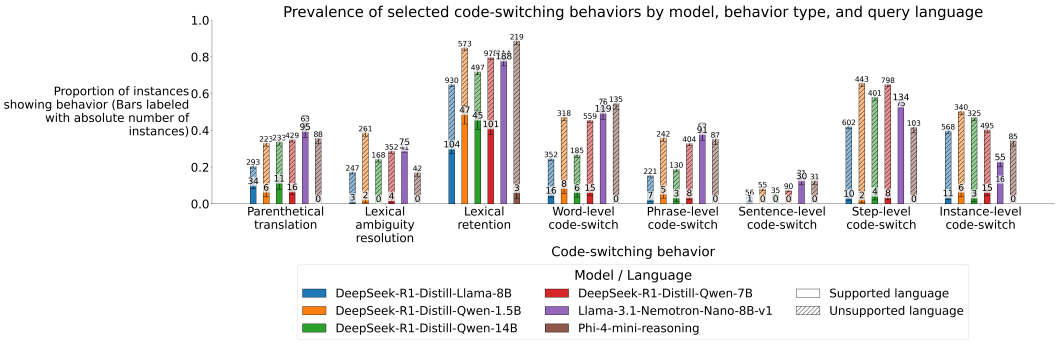

Figure 5: DeepSeek distilled, Llama-3.1-Nemotron-Nano-8B-v1, and Phi-4-mini-reasoning models code-switch more to handle queries in unsupported languages than for supported languages.

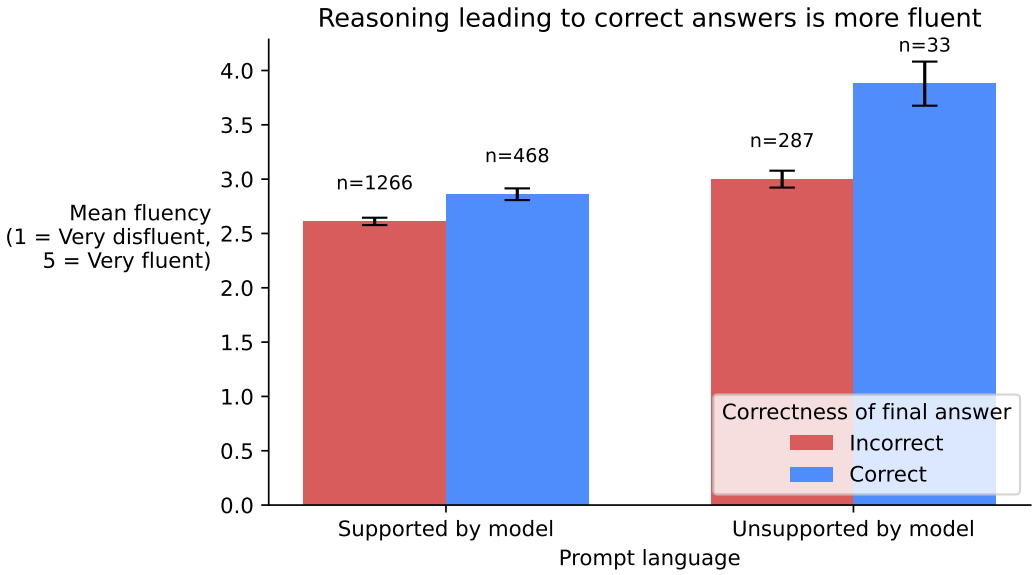

Figure 6: Our taxonomy includes a **Fluency** category that assesses the accuracy and naturalness of code-switching. For code-switched reasoning instances, reasoning that leads to correct answers is significantly more fluent than reasoning that leads to incorrect answers. Error bars show standard error of the mean.

NeurIPS paper checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly stated our claims, contributions, assumptions, and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a limitations section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed instructions for reproducing our approach, including which datasets, models, pipelines, and prompts we used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We link our data and code in the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide sufficient detail in the main body of the paper to understand the results, and additional details in the appendix and supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are included where suitable and the method for calculating the error bars is explained in the text of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include details of our compute resources in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: While our work presents minimal ethical risks, nonetheless we take appropriate preventative measures against potential harm, e.g., properly documenting and disclosing details of our experiments and methods for reproducibility and transparency.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a section on broader impacts of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release data or models that have a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the creators of relevant code, data, and models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

Justification: We provide proper documentation for our released code and data.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs in our Approach section.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.