Think in English, Answer in Korean: Efficient Adaptation of Multilingual Tool-Using Agents

Utsav Garg [*1] **Sungjin Hong** [*1] **Jason Jung** [*1]

Abstract

We present **LuckyStar 111B**[1], a 111B-parameter hybrid reasoning model developed through a collaboration between Cohere and LG CNS for Korean-English enterprise agents under practical memory and serving constraints. The model trains from Cohere's fully post-trained Command A (Cohere Team, 2025) model rather than a new pretraining run, and uses preamble conditioning to switch between concise non-reasoning behavior and longer tool-oriented reasoning. We study four choices for scaling tool-using agents efficiently: multilingual supervised fine-tuning, reinforcement learning with verifiable rewards for multi-step tool-use tasks, language-consistency rewards for Korean user-facing responses, and 4-bit quantization for single-GPU serving. The adapted model improves mathematical reasoning, function calling, and agentic natural-language-to-SQL (NL2SQL) performance while preserving general Korean and English instruction-following quality. These results provide a practical recipe and failure-mode analysis for adapting post-trained multilingual models to verifiable agentic workflows under memory-constrained deployment.

1. Introduction

Enterprise assistants must often reason over private data, call tools, and answer under tight memory and serving budgets. Multilingual use makes this harder. A Korean enterprise assistant, for example, may need to parse a Korean request, reason over English-heavy schemas or documentation, execute SQL or retrieval tools, and return a Korean answer whose numbers are grounded in the retrieved or computed result.

We study this setting with LuckyStar 111B, a 111B-parameter bilingual model jointly built by Cohere and LG CNS for Korean and English enterprise agents. Cohere has previously adapted enterprise models for specific languages and cultures, as in Command R7B Arabic (Alnumay et al., 2025); here we focus on adding hybrid reasoning and tool-use behavior in a bilingual Korean-English deployment setting. Instead of training a new model from scratch, we adapt a fully post-trained Command A (Cohere Team, 2025) model. This preserves general instruction-following and multilingual ability while reducing the cost and risk of adding specialized reasoning behavior. The work is motivated by a deployment setting in which agent quality, verifiable tool use, and memory footprint must be considered together.

The main design choice is *hybrid reasoning by preamble conditioning*. The same model weights support two operating modes. A reasoning preamble elicits longer step-by-step behavior for tool use, mathematics, and NL2SQL. A non-reasoning preamble elicits concise responses for ordinary user interactions. This avoids separate models for different enterprise workloads and lets downstream systems choose reasoning depth at inference time.

We make three contributions:

- We describe a three-stage adaptation pipeline for adding reasoning and tool-use behavior to a post-trained multilingual model without full retraining.
- We analyze multilingual reasoning failures, especially the tendency for Korean prompts to drift into English final answers, and introduce a reward penalty that improves language consistency during RLVR.
- We report deployment-oriented results showing that 4-bit quantization preserves benchmark quality in our evaluations while enabling single-H100 serving for a 111B-parameter model.

[*]Equal contribution; first authors are listed in alphabetical order. [1]Cohere. Correspondence to: SungJin Hong <sungjin-hong@cohere.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

[1]The name "LuckyStar" references the root of the LG brand, "Lucky Goldstar."

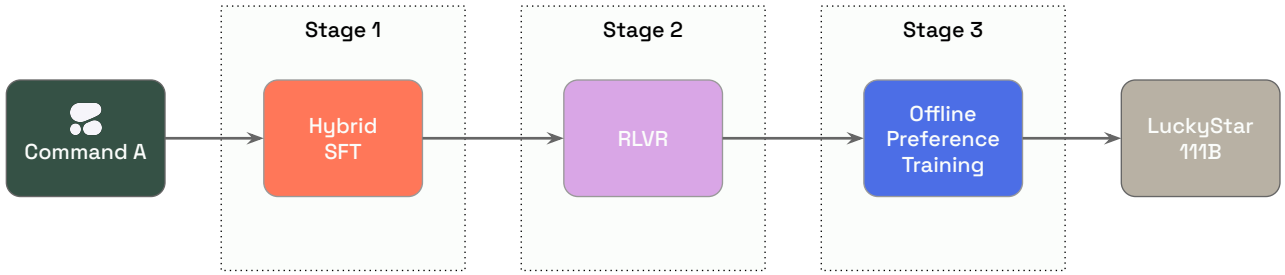

Figure 1. Training pipeline for LuckyStar 111B. A post-trained multilingual base model is adapted through hybrid SFT, RLVR for verifiable reasoning and tool use, and preference alignment for concise user-facing behavior.

2. Method

2.1. Hybrid Adaptation Pipeline

LuckyStar 111B is initialized from Command A (Cohere Team, 2025), a fully post-trained 111B-parameter enterprise model. We add reasoning ability through three stages: hybrid supervised fine-tuning (SFT), reinforcement learning with verifiable rewards (RLVR), and offline preference alignment using Direct Preference Optimization (DPO) (Ouyang et al., 2022; DeepSeek-AI et al., 2025; Rafailov et al., 2023).

Hybrid SFT teaches the two modes. Roughly 80% of the SFT mixture contains reasoning examples, while 20% preserves non-reasoning instruction-following behavior. The mix is biased toward reasoning because the base model already has strong multilingual instruction-following behavior. Reasoning examples use a preamble that encourages intermediate reasoning before the answer. Non-reasoning examples use a preamble that encourages concise final responses. At inference time, systems can select the desired behavior without changing weights or architecture.

The reasoning mixture includes math, code, and tool-use data. For math and code, we combine internal examples with public reasoning datasets (Guha et al., 2025; He et al., 2025). Some sourced problem sets contain prompts and final ground-truth answers but no solution traces. For these cases, we generate candidate step-by-step completions with strong internal and open models (Yang et al., 2025; DeepSeek-AI et al., 2025). We then extract the predicted final answer and retain only completions that exactly match the known ground truth. When answer extraction is unreliable, an LLM judge compares the predicted answer with the ground truth and checks whether the output follows the expected format. For a single prompt, multiple correct traces are often retained; this teaches the model that several reasoning paths can lead to the same answer and improves variation in reasoning style.

2.2. Multilingual Reasoning Strategy

Early experiments showed a consistent gap between English and Korean reasoning traces for the same Korean prompts. As shown in Table 1, English reasoning outperformed Korean reasoning even when the prompt was Korean. We attribute this to the English-heavy reasoning distribution and to tokenizer and reasoning-signal differences that favor English traces.

We first machine-translated 30% of English reasoning traces into Korean using an internal Command A configuration (command-a-03-2025, temperature 0.3, $p = 0.95$), with translations returned in JSON format for safe parsing. Thirty Korean annotators reviewed the translated samples to check naturalness and reasoning fidelity. This produced little improvement, likely because translation altered the naturalness of the reasoning traces and degraded the learning signal. We therefore used a mixed-language strategy: reason in English, but produce the final answer in the user's language. To teach this behavior, we sampled 20,000 Korean prompts with intermediate difficulty, generated English reasoning traces paired with Korean final answers through rejection sampling, and added verified examples to SFT.

2.3. Agentic NL2SQL Data

To support agentic reasoning for natural-language-to-SQL tasks, we curated NL2SQL data from Spider, BIRD, and SynSQL (Yu et al., 2018; Li et al., 2023; 2025). These datasets provide natural-language questions, gold SQL queries, and corresponding SQLite databases. Many raw examples were unsuitable for automated verification: some

Task	English Reasoning	Korean Reasoning
AIME 2024	69.3	50.9
MATH 500	93.6	85.6

Table 1. Pass@1 accuracy for Korean prompts when reasoning traces are generated in English or Korean during early SFT experiments.

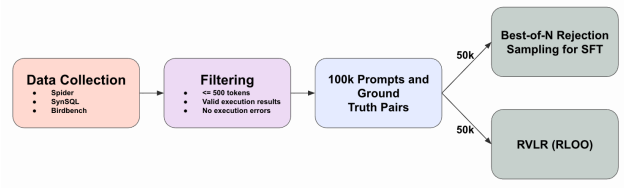

Figure 2. NL2SQL data preparation. Executable gold queries and verified model-generated alternatives provide rewardable tool-use examples.

gold SQL queries failed, some returned empty outputs, and others produced very long results. For RLVR and rejection sampling, we needed examples that could be checked automatically, so we retained only gold queries that executed successfully and returned outputs between one and 500 tokens. This yielded 100,000 verifiable prompt-query pairs.

Enterprise tools are a natural setting for verifiable rewards because many actions produce structured or executable outputs. SQL execution, retrieval-grounded answer checks, and formatting constraints can provide sharper training signals than preference-only supervision for workflows where correctness is operationally important.

The base policy achieved less than 1% accuracy on this curated NL2SQL set, creating a cold-start problem with insufficient reward signal for reinforcement learning. We first bootstrapped the policy with SFT examples generated by best-of-N rejection sampling (Verdun et al., 2025). A subset of prompts was held out from the RLVR stage to preserve starting entropy. A high-performing internal Cohere model and open models received a read-only SQL execution tool and generated eight candidate solutions per prompt. Candidate queries were executed and verified with a two-stage pipeline combining heuristic checks and an LLM judge for ambiguous cases. We retained all verified-correct traces, including multiple diverse solutions for the same prompt, then integrated the resulting agentic NL2SQL examples into the hybrid SFT mixture.

2.4. RLVR and Language Consistency

The RLVR stage optimizes reasoning and tool use with binary rewards. We use REINFORCE Leave-One-Out (RLOO) (Ahmadian et al., 2024) and custom reward functions for mathematics and agentic NL2SQL. The training data contains over 50,000 agentic NL2SQL prompts and 4,000 mathematics prompts. For each task, the final answer is evaluated against a ground truth using a rule-based verifier when possible. When deterministic rules are insufficient, an LLM judge based on Command A assigns a binary reward. To optimize training, we select prompts whose pre-RLVR pass rate is between 12.5% and 87.5%, avoiding examples that are either solved too easily or provide no reward signal.

Training Step	Language Mismatch Rate
Step 0 (SFT)	12.2%
Step 100	3.3%
Step 200	0.8%

Table 2. Final answer language mismatch rate for Korean prompts during RLVR.

For Korean prompts, early RLVR runs often optimized toward English final answers. We therefore use a language-consistency penalty:

$$r = r_{\text{correctness}} - r_{\text{penalty}},$$
$$r_{\text{penalty}} = \begin{cases} 0.5 & \text{if final answer language mismatches,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Prompt languages are pre-classified, and ambiguous prompts are excluded. An LLM judge checks whether the final user-facing response matches the prompt language while the intermediate reasoning remains in English. As shown in Table 2, this reduces Korean-to-English answer drift during RLVR.

To avoid rewarding premature termination, rollouts exceeding 32k tokens are filtered from training batches rather than assigned zero reward. This choice follows the observation that complex reasoning can require long chains of thought, so a zero reward for length can incorrectly teach the model to stop early. RLVR led to improvements in reasoning and agentic NL2SQL, and we also observed fewer repetition-related failures. However, this improvement came at a cost to conciseness: after RLVR, the model frequently failed to follow instructions requiring brief responses, especially in multiple-choice settings where it would elaborate instead of answering directly.

2.5. Offline Preference Alignment

To correct the verbosity introduced during RLVR, we apply low-learning-rate DPO over preference pairs (Rafailov et al., 2023). The preference set combines data from the base model's alignment pipeline with newly curated Korean-specific preference data. We also generate targeted preference pairs for conciseness: direct answers are labeled as chosen, while verbose post-RLVR outputs are labeled as rejected. The low learning rate is used to refine chatbot behavior without compromising reasoning and tool-use capabilities acquired during RLVR.

	Mathematical Reasoning				Tool Use		
	Korean		English		Enterprise NL2SQL	BFCL v3	LG Agentic Eval
	AIME 2024	MATH 500	AIME 2024	MATH 500			
LuckyStar 111B	69.3	93.6	73.7	94.0	38.0	70.6	4.85
LuckyStar 111B 4-bit[2]	66.0	95.6	72.3	96.0	–	–	–
Command A	10.0	76.2	13.3	79.6	7.3	64.4	2.67
GPT-4o (11/20)	16.7	76.4	13.3	78.6	37.3	71.7	2.56
Claude 3.7 Sonnet	46.7	92.8	50.0	92.5	38.0	58.6	5.04
Qwen3 235B A22B	80.0	92.4	80.0	91.2	22.0	67.9	3.29

Table 3. Reasoning and tool-use results. Mathematical tasks report Pass@1 accuracy.

3. Results

3.1. Reasoning and Tool Use

Table 3 compares LuckyStar 111B against the base model and strong proprietary and open models, including GPT-4o (OpenAI, 2024), Claude 3.7 Sonnet (Anthropic, 2025), and Qwen3 235B A22B (Yang et al., 2025). LuckyStar 111B improves over Command A on Korean and English mathematical reasoning, BFCL v3 function calling (Patil et al., 2025), and internal enterprise tool-use evaluations. For mathematical reasoning, we evaluate AIME 2024 (Mathematical Association of America, 2024) and MATH 500 (Hendrycks et al., 2021; Lightman et al., 2023) in English and Korean-translated form, using a 32k-token maximum context window for all models.

All proprietary-model baselines reflect the evaluation snapshot available at the time these experiments were run. We did not re-run newer model releases after the evaluation freeze.

We evaluate agentic and tool-calling capabilities using public and internal benchmarks. BFCL v3 measures function-calling behavior. Enterprise NL2SQL is an internal Cohere 150-question NL2SQL benchmark spanning multiple business domains. The LG Agentic Evaluation, developed by LG CNS, contains 60 business and finance questions requiring retrieval, SQL, or both; outputs are scored for numerical accuracy, formatting, and intent satisfaction. We include internal enterprise evaluations because public tool-use benchmarks do not fully capture the workflows, schema and formatting requirements, and numerical accuracy constraints that determine whether an agent is useful in deployment. These targeted benchmarks provide strong development signal for fast iteration and for scaling model capabilities toward the final enterprise goal, while public benchmarks provide comparability.

LuckyStar 111B improves Enterprise NL2SQL from 7.3 to 38.0 and LG Agentic Evaluation from 2.67 to 4.85. The gains over Command A are large, but the absolute scores show that realistic enterprise tool use remains difficult.

[2]Tool-use results are not reported for the 4-bit model because the evaluation runs were not completed before the final evaluation cutoff.

3.2. General Multilingual Quality

Table 4 shows that the specialized reasoning pipeline preserves most general instruction-following and multilingual quality. LuckyStar 111B maintains strong Korean and English performance on knowledge, reasoning, and instruction-following benchmarks. Korean ARC, IFEval, and MT-Bench are translated versions of the English evaluations. On Korean benchmarks, LuckyStar 111B reaches 68.6 on KMMLU, 89.2 on ARC Challenge, 77.8 on IFEval, and 8.09 on MT-Bench. In English, it reaches 82.7 on MMLU, 93.8 on ARC Challenge, 89.4 on IFEval, and 8.45 on MT-Bench. These results indicate that the reasoning and tool-use gains do not come at the cost of broad Korean and English instruction-following quality.

3.3. Efficient Deployment

Enterprise deployments are constrained by inference cost, memory, and available GPU capacity. We quantize LuckyStar 111B from FP8 to 4-bit weights, reducing memory footprint by approximately 50% and lowering memory-bandwidth requirements. This enables the 111B-parameter model to run on a single 80GB H100 GPU, lowering the barrier to deployment in enterprise settings. As shown in Tables 3 and 4, the 4-bit model closely matches the FP8 model across the reported reasoning and academic benchmarks. We view this result as a deployment feasibility check rather than a complete serving study; detailed latency, throughput, and energy measurements remain important future work.

4. Discussion

These results suggest that multilingual enterprise agents can benefit from separating the internal reasoning language from the user-facing response language. In our experiments, English reasoning traces provide stronger learning signals for verifiable reasoning, while language-consistency rewards keep Korean prompts from producing English final answers. This design is practical but imperfect: native Korean rea-

	Korean				English			
	KMMLU	ARC-C	IFEval	MT-Bench	MMLU	ARC-C	IFEval	MT-Bench
LuckyStar 111B	68.6	89.2	77.8	8.09	82.7	93.8	89.4	8.45
LuckyStar 111B 4-bit	67.9	88.6	78.3	8.06	82.1	92.8	89.3	8.50
Command A	64.9	88.8	77.2	8.18	83.9	93.4	90.9	8.34
GPT-4o	66.3	90.5	70.2	8.73	83.7	91.8	83.0	8.53
Claude 3.7 Sonnet	67.9	89.0	74.9	8.81	84.6	95.0	87.9	8.47
Qwen3 235B A22B	67.0	90.3	71.3	8.87	83.4	93.2	86.4	8.73

Table 4. General Korean and English benchmark performance. KMMLU and MMLU evaluate knowledge; ARC-C evaluates science reasoning; IFEval measures instruction following; MT-Bench is a conversation-quality judge benchmark (Son et al., 2024; Hendrycks et al., 2020; Clark et al., 2018; Zhou et al., 2023; Zheng et al., 2023).

soning remains a target for future work, especially for tasks where cultural or domain-specific knowledge may be better expressed in Korean.

The NL2SQL experiments also show the importance of cold-start data for tool-use RL. In the curated NL2SQL set, the base model produces too few correct traces for RLVR alone. Rejection-sampled SFT examples create enough initial competence for RL to improve policy. This is consistent with the broader observation that reward quality and prompt selection are central to making verifiable RL useful for agentic tasks.

During development, three failure modes were especially important: NL2SQL cold-start failures, language drift from Korean to English during RLVR, and verbosity after reasoning optimization. The final pipeline addresses these with rejection-sampled SFT, a language-consistency reward, and preference alignment. We believe these failure modes are broadly relevant to scalable agentic systems because they arise at the interface between verifiable rewards, multilingual behavior, and deployment-oriented model adaptation.

More broadly, our results support a deployment-scoped view of scaling: practical agent systems do not always require a single model optimized for every capability. For many enterprise settings, it can be more useful to adapt an existing post-trained model toward a focused set of verifiable workflows, while preserving general instruction-following behavior and reducing serving requirements through quantization.

This work has several limitations. Some evaluations are internal because they reflect deployment-specific workflows, schema conventions, and formatting requirements that are not fully captured by public benchmarks; these benchmarks are useful for development but should complement, not replace, public evaluations. Details about the proprietary data and internal evaluation sets are necessarily limited due to legal, privacy, and confidentiality obligations, which also limits reproducibility. Finally, the model still relies on English reasoning for many Korean prompts, leaving native Korean reasoning as an important direction for future work.

5. Conclusion

We presented LuckyStar 111B, a bilingual tool-using agent model adapted from a post-trained 111B-parameter base. Preamble conditioning lets one model support reasoning and non-reasoning modes. Verifiable NL2SQL rewards, language-consistency penalties, and preference alignment improve tool use while preserving Korean and English instruction-following behavior. Finally, 4-bit quantization makes the model deployable on a single H100 with little observed quality loss in the reported evaluations. Future work will focus on expanding native-language reasoning data, improving Korean domain adaptation, measuring serving efficiency more comprehensively, and integrating the model more closely with enterprise knowledge bases and tools.

Acknowledgments

Secondary authors: **Justin Lee, Shaan Desai, Joon Hee Kim, Anirudh Shrinivason, Edmond Wen, and Susie Park.**

This work was a collaboration between many teams at Cohere and LG CNS. We particularly appreciate the members of the GenAI Product Lab and Agentic AI Lab from LG CNS AI Lab. We also acknowledge the following people at Cohere who supported the project: Andrew Chang, Eugene Cho, Q Cho, Elliott Choi, Manoj Govindassamy, Vi Iyengar, Edward Kim, Jiyeon Lee, Jeffrey Li, Jonathan Li, Olivia Markham, Adrien Morisot, Vivek Muppalla, Jeremy Pekmez, Max Pfeifer, Sudip Roy, Michael Sachs, Isha Satyakam, Tom Sherborne, and Ali Edalati.

LLM/Agent Usage Disclosure

The research described in this paper used LLMs as part of the model-development pipeline: generating candidate reasoning traces, judging answer equivalence for examples where deterministic verification was insufficient, checking final answer language consistency, and producing candidate SQL traces with a read-only execution tool. LLM was used to help proof-read the paper to improve clarity and flow. The authors are responsible for all technical claims, experiments, and the final text.

References

Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

Alnumay, Y., Barbet, A., Bialas, A., Darling, W., Desai, S., Devassy, J., Duffy, K., Howe, S., Lasche, O., Lee, J., Shrinivason, A., and Tracey, J. Command R7B arabic: A small, enterprise focused, multilingual, and culturally aware arabic LLM, 2025.

Anthropic. Claude 3.7 sonnet, 2025.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cohere Team. Command A: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*, 2025. doi: 10.48550/arXiv.2504.00698.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., and Gao, Z. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025.

Guha, E., Marten, R., Keh, S., Raoof, N., Smyrnis, G., Bansal, H., Nezhurina, M., Mercat, J., Vu, T., Sprague, Z., Suvarna, A., Feuer, B., Chen, L., Khan, Z., Frankel, E., Grover, S., Choi, C., Muennighoff, N., Su, S., Zhao, W., Yang, J., Pimpalgaonkar, S., Sharma, K., Ji, C. C.-J., Deng, Y., Pratt, S., Ramanujan, V., Saad-Falcon, J., Li, J., Dave, A., Albalak, A., Arora, K., Wulfe, B., Hegde, C., Durrett, G., Oh, S., Bansal, M., Gabriel, S., Grover, A., Chang, K.-W., Shankar, V., Gokaslan, A., Merrill, M. A., Hashimoto, T., Choi, Y., Jitsev, J., Heckel, R., Sathiamoorthy, M., Dimakis, A. G., and Schmidt, L. Openthoughts: Data recipes for reasoning models, 2025.

He, Z., Liang, T., Xu, J., Liu, Q., Chen, X., Wang, Y., Song, L., Yu, D., Liang, Z., Wang, W., Zhang, Z., Wang, R., Tu, Z., Mi, H., and Yu, D. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*, 2021.

Li, H., Wu, S., Zhang, X., Huang, X., Zhang, J., Jiang, F., Wang, S., Zhang, T., Chen, J., Shi, R., et al. Omnisql: Synthesizing high-quality text-to-SQL data at scale. *arXiv preprint arXiv:2503.02240*, 2025.

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., et al. Can LLM already serve as a database interface? a big bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42330–42357, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Mathematical Association of America. American invitational mathematics examination - AIME 2024. In *American Invitational Mathematics Examination - AIME 2024*, February 2024.

OpenAI. GPT-4o system card, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Patil, S. G., Mao, H., Ji, C. C.-J., Yan, F., Suresh, V., Stoica, I., and Gonzalez, J. E. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Son, G., Lee, H., Kim, S., Kim, S., Muennighoff, N., Choi, T., Park, C., Yoo, K. M., and Biderman, S. KMMLU: Measuring massive multitask language understanding in korean. *arXiv preprint arXiv:2402.11548*, 2024.

Verdun, C. M., Oesterling, A., Lakkaraju, H., and Calmon, F. P. Soft best-of-n sampling for model alignment, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025.

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. *arXiv preprint arXiv:1809.08887*, 2018.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-Bench and chatbot arena, 2023.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.