TRAINING DOMAIN DRAFT MODELS FOR SPECULATIVE DECODING: BEST PRACTICES AND INSIGHTS

Fenglu Hong, Ravi Raju, Jonathan Lingjie Li, Bo Li, Urmish Thakker,
Avinash Ravichandran, Swayambhoo Jain, Changran Hu
SambaNova Systems, Inc.
Palo Alto, CA, USA
{fenglu.hong,ravi.raju,jonathan.li,bo.li,urmish.thakker,
avinash.ravichandran,swayambhoo.jain,changran.hu}@sambanovasystems.com

ABSTRACT

Speculative decoding is an effective method for accelerating inference of large language models (LLMs) by employing a small draft model to predict the output of a target model. However, when adapting speculative decoding to **domain-specific** target models, the acceptance rate of the **generic** draft model drops significantly due to domain shift. In this work, we systematically investigate knowledge distillation techniques for **training domain draft models** to improve their speculation accuracy. We compare white-box and black-box distillation approaches and explore their effectiveness in various data accessibility scenarios, including historical user queries, curated domain data, and synthetically generated alignment data. Our experiments across *Function Calling*, *Biology*, and *Chinese* domains show that offline distillation consistently outperforms online distillation by 11% to 25%, white-box distillation surpasses black-box distillation by 2% to 10%, and data scaling trends hold across domains. Additionally, we find that synthetic data can effectively align draft models and achieve 80% to 93% of the performance of training on historical user queries. These findings provide practical guidelines for training domain-specific draft models to improve speculative decoding efficiency.

1 INTRODUCTION

Large language models (LLMs) like GPT-4 (OpenAI et al., 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025) have demonstrated remarkable capabilities, but come with high computational costs and inference latency, limiting real-time applications. As a solution, speculative decoding accelerates model inference by using a smaller model (known as the draft model) to generate candidate outputs, which are then verified by the larger model (known as the target model) (Leviathan et al., 2023; Chen et al., 2023). However, the performance of speculative decoding heavily depends on draft-target model alignment — misalignment increases verification failures, requiring the draft model to regenerate tokens. Our experiments reveal that, when replacing a generic target model with a domain-specific fine-tuned model in speculative decoding, the speculation accuracy of the draft model in terms of the average token acceptance rate drops significantly in domain-specific queries, as shown in Table 1. This degradation underscores the need for domain-adapted draft models to maintain efficiency in speculative decoding.

	Generic Target	Domain Target	Relative Drop
Biology	60.7	37.5	-38.2%
Chinese	49.6	35.7	-28.0%
Coding	59.4	55.2	-7.1%
Math	78.3	75.4	-3.7%

Table 1: Avg. token acceptance rate (%) for general target model and domain target model, using a general small model as draft model. Details in Appendix C.1.

To improve speculative decoding efficiency for domain target models, we explore various knowledge distillation techniques for training domain draft models (Agarwal et al., 2024; Hinton et al., 2015; Zhou et al., 2024) under various data constraints. We compare white-box distillation, which utilizes the target model parameters, and black-box distillation, which

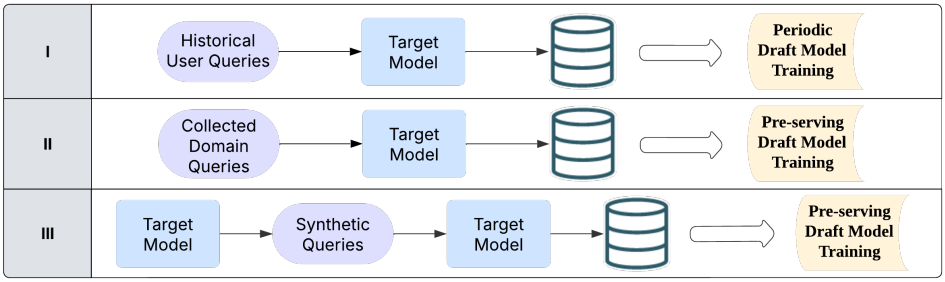

Figure 1: Three data accessibility scenarios for domain draft model training. Scenario I assumes access to historical user queries and train the draft model with distillation losses given target model's generations. Scenario II and III assume no access to use queries. We can use either collected domain queries (II) or synthetic queries generated by the target model (III) for training.

relies only on the target model outputs. We also examine training strategies under varying data availability conditions. Ideally, training with historical user queries ensures minimal domain shift and optimal performance. However, in many real-world scenarios, such data may not be available, particularly before a model has seen real user interaction (Liu et al., 2024). To address this, we evaluated alternatives, including training with curated domain queries and using synthetically generated data, such as Magpie (Xu et al., 2024), from the target model. Our work provides practical guidelines for training draft models under different constraints and thus improving speculative decoding efficiency in domain-specific applications.

In summary, our paper makes the following contributions:

- Comprehensive analysis of distillation methods: We compare white-box and black-box distillation for training draft models and evaluate their relative effectiveness across different domains.
- Investigation of data accessibility constraints: We explore training strategies under three different data scenarios: (1) historical streaming/user query data, (2) collected domain-specific queries, and (3) synthetically generated data.
- Empirical evaluation across multiple domains: We conduct experiments on three target domains (Function Calling, Biology, and Chinese) to assess the impact of different training methods on speculative decoding performance.
- Guidance for practical draft model training: Our findings provide insights on how to construct draft models under varying data constraints, offering a practical reference for improving inference efficiency in domain-specific LLM applications.

2 Background & Methods

2.1 Knowledge Distillation (KD) for speculative decoding

The effectiveness of speculative decoding depends on how well the output distribution from the draft model aligns with that of the target model. Knowledge distillation, which is a widely used framework for training a smaller student model to mimic the predictive distribution of a larger teacher model, is thus effective for enhancing speculative decoding (Zhou et al., 2024; Liu et al., 2024). To develop more effective domain-specific draft models for speculative decoding, we use knowledge distillation techniques to improve the alignment between the target model (M_p) and draft model (M_q). We assume that the draft has learnable parameters θ. We are also given a dataset of input sequences X. We generate **outputs from the target model** with greedy decoding, to form the training dataset $D = \{(x_k, p(x_k))\}_{k=1}^{|X|}$.

Supervised FT on target model's outputs The draft model is finetuned to minimize the negative log-likelihood L_{SFT} over the output sequences from the target model. In subsequent sections, this method is referred to as **SFT**.

$$L_{SFT}(\theta) := \mathbb{E}_{(x,y)\sim(X,Y)} \left[-\log q_\theta(y|x) \right] \tag{1}$$

Supervised KD on target model's outputs This is a white-box distillation technique where the draft model is trained to mimic the token-level probability distributions of the target model. Specifically, the draft model is trained with

$$L_{KD}(\theta) := \mathbb{E}_{(x,y)\sim(X,Y)} \left[\mathcal{D}\left(p \| q_\theta\right)(y|x) \right], \tag{2}$$

where we use the forward Kullback Leibler divergence (KL) and reverse KL (RKL) for $\mathcal{D}$.

Online vs. Offline Distillation For the white-box distillation, we further investigate two learning paradigms: online distillation and offline distillation, as proposed in (Liu et al., 2024). In offline distillation setting, the draft model has unrestricted access to the static dataset D. Offline distillation does not allow real-time adaptation to shifts in data distribution. In contrast, online distillation refines the draft model dynamically during the speculative decoding inference process. Specifically, the draft model proposes tokens during inference, which are verified by the target model. The target logits and draft logits for the incorrect predictions are stored in a buffer, and the draft model is updated every time when the buffer exceeds a threshold.

2.2 MAGPIE ALIGNMENT DATA SYNTHESIS

Magpie is a synthetic data generation technique used to generate training data for model alignment (Xu et al., 2024). By providing the (aligned) target model with a pre-query template (and potentially a system prompt), the model generates both sample queries and corresponding completions. We re-purpose Magpie for creating draft model training data for speculative decoding. This approach offers two key advantages: (1) it is data-free, requiring no pre-existing datasets, and (2) it eliminates biases associated with selecting training data.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We adopt the setup presented in *Online Speculative Decoding* paper (Liu et al., 2024) for white-box distillation and evaluate our methods primarily on the LLaMA series (Grattafiori et al., 2024). Specifically, we use the LLaMA 3.2 1B Instruct model[1] as the draft model and, unless otherwise specified, conduct all experiments with LLaMA 3.1 8B-sized target models. To assess the effectiveness of distillation approaches, we focus on adapting the draft model to niche domains; we select 1) Function Calling, 2) Biology, and 3) Chinese to ensure sufficient coverage. For each domain, we select a domain-specific target model and use open-source domain datasets from Hugging face, setting aside 1000 prompts for the test set. To evaluate performance, we measure the *average token acceptance rate* of the draft model on the test set with proposal length $k = 9$. More details on target models, datasets, and training configurations are discussed in Appendix B.

3.2 DATA ACCESSIBILITY SCENARIOS

We study three different data accessibility scenarios and corresponding data collection techniques for training domain-specific draft models, as depicted in Figure 1.

I. Using historical user query data This scenario assumes minimal domain shift between training and deployment. To simulate such a setting, we select a domain-specific dataset D and apply a train-test split for training the draft model and evaluating it under speculative decoding setting.

II. Using collected domain-specific queries To simulate scenario where the user query data is unavailable but related domain queries exist, we train the draft model on dataset D' and evaluate it on a separate dataset D. Both datasets belong to the same domain, but their queries may exhibit minor domain shifts, providing insight into the robustness of draft model training under distributional variations.

[1] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

III. Using synthetically generated queries Collecting domain queries often requires significant human effort. However, when user queries and curated domain queries are unavailable, we leverage the target model to generate synthetic instructions and corresponding completions. Specifically, we adopt the synthetic data generation method proposed by (Xu et al., 2024), which eliminates the need for prompt engineering or manually curated seed instructions.

4 Results and Discussion

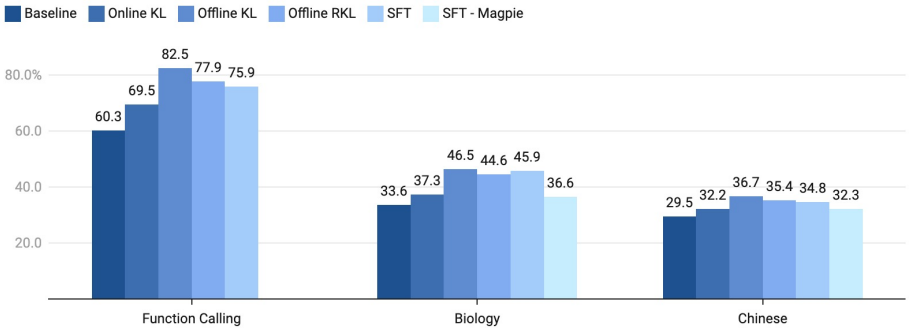

Figure 2: Average acceptance rates for different methods. All methods (except SFT - Magpie) train with in-domain data where a domain-specific dataset is split into training and test sets, mimicking real user queries (Scenario I). SFT - Magpie method trains with Magpie synthetic data (Scenario III). More details in Appendix C.2.

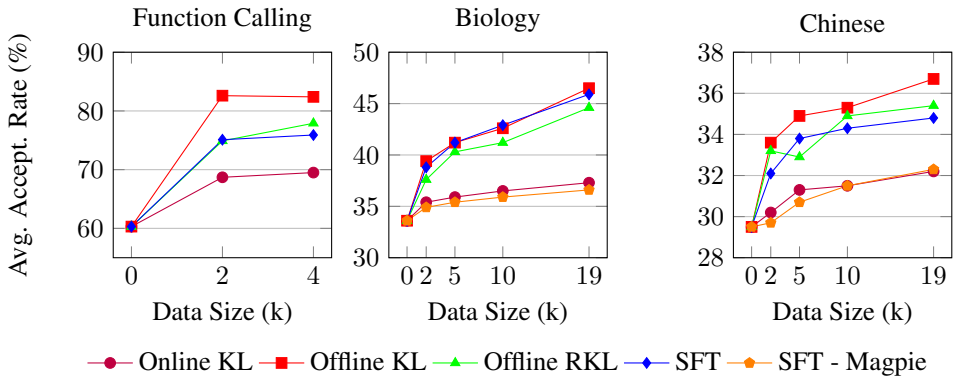

Figure 3: Performance scales with dataset size. Besides, as training data increases, the offline KL approach gains an increasing advantage over online KL in Biology and Chinese domains.

In this section, we present our experimental findings and analyze the impact of different distillation methods on speculative decoding performance across various domains.

Offline vs. Online Distillation Our experiments show that offline distillation consistently outperforms online distillation across all three domains (see Figure 2), and this trend holds across different dataset sizes (see Figure 3). A key insight into this trend is that offline distillation leverages supervision from all completion tokens, providing richer learning signals for the draft model. Furthermore, as training data increases, the offline approach gains a growing advantage over online approach in Biology and Chinese. In Biology, offline surpasses online by 11.4% to 24.7% in acceptance rate as data expands from 2k to 19k; in Chinese, the gap widens from 11.2% to 14.1%. We also find out that for training with in-domain user queries data (Data Scenario I), offline distillation can benefit from higher learning rate while online distillation requires a lower learning rate, as detailed in Appendix D.1. However, for Data Scenario II and III where the training data exhibits domain shifts from evaluation data, offline distillation would also prefer a lower learning rate (see Table 4 and 5).

[2] https://huggingface.co/aaditya/Llama3-OpenBioLLM-70B
[3] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct

Target Model	Draft Model	Biology
Llama3-OpenBioLLM-70B[2]	Llama3.2-1B[3]	33.0%
	+ SFT (Data Scenario I)	+8.4%

Table 2: Accept. rate(%) of 1B **SFT**-trained draft model for decoding 70B target model in Biology domain. Training data size is 4k.

Target Model	Draft Model	Math	Coding	General	Avg.
DS-R1-Qwen-32B[4]	Qwen2.5-0.5B[5]	34.7%	35.6%	34.6%	35.0%
	+ SFT (Data Scenario II)	+5.0%	+5.0%	+5.0%	+5.0%
DS-R1-Llama-70B[6]	Llama3.2-1B[7]	41.6%	32.3%	26.9%	33.6%
	+ SFT (Data Scenario II)	+10.0%	+17.7%	+6.0%	+11.2%

Table 3: Accept. rate(%) of **SFT**-trained draft models for decoding 70B and 32B target models in Reasoning domain. Training data size is 200k.

Data Scaling Law As shown in Figure 3, we observe that the data scaling law generally holds across all domains and training methods, with larger datasets yielding better draft model alignment. One exception occurs in Function Calling domain with offline distillation using forward KL loss, where 2000 training samples already reach optimal performance. This is likely due to the highly structured nature of function-calling outputs, which require less training data to achieve alignment.

White-Box vs. Black-Box Distillation White-box offline distillation with forward KL loss generally outperforms black-box distillation (SFT) (see Figure 2 and 3), indicating that leveraging the target model's logits provides a stronger training signal than relying solely on final output tokens. In our investigated domains, the former achieves $1.3\% \sim 9.9\%$ more acceptance rate than SFT.

Effectiveness of Synthetic Data (Magpie) We examine the effectiveness of Magpie synthetic data for aligning draft models in Biology and Chinese domains. While not as effective as training on in-domain data, synthetic data still yields meaningful improvements (see Figure 2), making it a viable approach for training an initial draft model before target model deployment. Notably, the performance gap between user query finetuning and synthetic data finetuning is larger in Biology than in Chinese. This can be attributed to the fact that the biomedical target model primarily generating diagnostic and medical-related completions, which exhibit a greater domain shift from real user queries that span biological topics. We also find that offline training with Magpie data benefits from a smaller learning rate than in-domain user query data and achieves over 90% of the acceptance rate observed with in-domain training (see Table 4). This is likely because Magpie data exhibits some domain shift from the evaluation data, and thus a larger LR leads to overfitting.

Scaling to Larger Target Model In order to assess the effectiveness of training draft models for larger target models, we extend our experiments to models exceeding 8B parameters in the domains of Biology and Reasoning. As presented in Table 2, within the Biology domain, SFT with 4,000 training samples improves the acceptance rate of the 1B draft model by 25.4% for the 70B target model (Llama3-OpenBioLLM-70B).

Additionally, we evaluate our method for two reasoning models distilled from DeepSeek-R1 (DeepSeek-AI et al., 2025), which incorporate extended reasoning processes in their outputs and demonstrate strong performance in reasoning-intensive tasks. To conduct our experiments within the framework of Data Scenario II, we use a multi-domain instruction dataset for training and reasoning-intensive domain datasets for evaluation. Specifically, we sample 200,000 prompts from a combination of the training dataset introduced in Jain et al. (2024) and the Magpie-500K dataset[8], covering math, coding, and additional domains. For evaluation, we construct domain-specific datasets for

[4] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B&qwen2.5-0.5B-Instruct
[5] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct
[6] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
[7] https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
[8] https://huggingface.co/datasets/Magpie-Align/Magpie-Llama-3.1-Pro-500K-Filtered

	Data Size	LR=1e-6	LR=2e-5
Offline KL on Chinese train set (Data Scenario I)	19k	35.6	**36.7**
Offline KL on Magpie syn. data (Data Scenario III)	19k	**33.2**	31.7

Table 4: Avg. accept. rate (%) on Chinese test set. The baseline accept. rate is 29.5%. Offline KL with Magpie synthetic data achieves 90.4% of the performance of training with in-domain data.

	Data Size	LR=1e-6	LR=2e-5
Offline KL on Hermes-FC[11] (Data Scenario I)	2k	76.4	**82.6**
Offline KL on APIGen-FC[12] (Data Scenario II)	20k	**72.1**	63.6

Table 5: Avg. accept. rate(%) on the test split of Hermes-FC dataset. The baseline accept. rate is 60.3%. Training on related domain data reaches 87.3% of the performance of using in-domain data.

math, *coding*, and *general* reasoning by sampling 500 prompts from AIME[9], BigCodeBench[10], and AlpacaEval (Dubois et al., 2024), respectively. As summarized in Table 3, the SFT-trained Qwen 2.5 0.5B Instruct model achieves a 14.3% improvement in acceptance rate when serving as a draft model for the 32B reasoning model. Similarly, the SFT-trained LLaMA 3.2 1B Instruct model exhibits a 33.3% improvement when used as a draft model for the 70B reasoning model. These results indicate that the effectiveness of SFT extends to larger target models, and training on collected domain-relevant data can significantly enhance the alignment of draft models.

Training on Related Domain Data For Data Scenario II, when user queries are unavailable, training on related domain data also effectively improves draft model alignment. In Function Calling domain, training on APIGen data[13] enhances the draft model's performance on Hermes-FC[14] (Table 5), which serves as a proxy for real user queries. However, domain shift between collected data and user queries necessitates significantly more training samples to achieve comparable results and more careful learning rate selection. Beyond the Function Calling domain, as discussed in the previous section, SFT on collected domain datasets also greatly improves the performance of the draft models in math, coding, and general reasoning tasks (see Table 3).

5 Conclusion

This work investigates best practices for training domain-specific draft models to improve speculative decoding efficiency when paired with specialized target models. Our experiments show that offline knowledge distillation outperforms online learning by 11% to 25%, with forward KL loss providing the optimal result. We also demonstrate that white-box distillation, which utilizes target model logits, exceeds the black-box approach by 2% to 10%. Additionally, we explore data accessibility scenarios and find that synthetic alignment data can achieve 80% to 93% of the performance of training on in-domain data. These insights provide actionable guidelines for the construction of effective draft models under different constraints, ultimately enhancing speculative decoding for domain-specific applications.

[9] https://huggingface.co/datasets/di-zhang-fdu/AIME_1983_2024
[10] https://huggingface.co/datasets/bigcode/bigcodebench
[11] https://huggingface.co/datasets/NousResearch/hermes-function-calling-v1
[12] https://huggingface.co/datasets/argilla/apigen-function-calling
[13] https://huggingface.co/datasets/argilla/apigen-function-calling
[14] https://huggingface.co/datasets/NousResearch/hermes-function-calling-v1

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024. URL https://arxiv.org/abs/2305.14387.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu,

Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent,

Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2306.08543.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Swayambhoo Jain, Ravi Raju, Bo Li, Zoltan Csaki, Jonathan Li, Kaizhao Liang, Guoyao Feng, Urmish Thakkar, Anand Sampat, Raghu Prabhakar, and Sumati Jairath. Composition of experts: A modular compound ai system leveraging large language models, 2024. URL https://arxiv.org/abs/2412.01868.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding, 2024. URL https://arxiv.org/abs/2310.07177.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez,

Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605/.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing, 2024. URL https://arxiv.org/abs/2406.08464.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation, 2024. URL https://arxiv.org/abs/2310.08461.

A Related Work

A.1 Speculative Decoding for LLMs

Speculative decoding (SD) is a means to accelerate LLM decoding by leveraging a small draft model to predict the outputs from a large target model. These candidate output tokens are then validated by a larger target model (Leviathan et al., 2023; Chen et al., 2023). This decoupling of candidate generation from verification permits a reduction in the number of target model invocations, thereby decreasing overall inference time.

Most prior work in speculative decoding focuses on accelerating general-purpose LLMs by pairing them with small, general-purpose draft models—even when evaluations are performed on domain-specific tasks. In contrast, our study systematically investigates how to employ knowledge distillation to adapt a general-purpose draft model for use with a domain-specific target model. This work provides practical guidelines and best practices for aligning the draft model to the specialized output distribution, thereby reducing inference latency while preserving domain-specific performance.

A.2 Knowledge Distillation for LLMs

Knowledge distillation (KD) has long been recognized as an effective strategy for compressing large models into smaller, more efficient ones without a substantial loss in performance (Hinton et al., 2015). In the context of LLMs, KD enables a smaller "student" model to learn from a larger "teacher" model, thereby significantly reducing inference cost while striving to maintain high-quality output generation.

Early applications of KD in LLMs have primarily focused on black-box distillation (Taori et al., 2023), where the student model is trained solely on the teacher's outputs, often accessed via APIs. While practical when teacher internals are unavailable, this approach inherently limits the granularity of supervision. Recent advances have introduced white-box KD (Zhou et al., 2024; Agarwal et al., 2024; Gu et al., 2024; Wen et al., 2023), which leverages internal representations such as logits and attention maps from the teacher to provide richer supervisory signals and, consequently, improved student performance.

B Experiment Details

In this section, we provide detailed information on our experiment setup.

B.1 Domain datasets and target models

This section includes information about the target model selection and dataset selection for training and evaluation.

We select one Llama 3.1 8B-sized target model for each domain that we study. Specifically, we use Hermes-3-Llama-3.1-8B model for Function Calling[15], the Llama-3.1-8B-UltraMedical model for the Biology domain[16], and the Llama3.1-8B-Chinese-Chat for the Chinese domain[17].

For the domain-specific datasets that are used to mimic real user queries (Scenario I), we make the following selection:

- Function Calling: We use *NousResearch/hermes-function-calling-v1*[18] dataset (Hermes-FC) and exclude Subset "function_calling", which consists of multi-turn conversation function calls.
- Biology: We use *camel-ai/biology*[19] dataset.

[15] https://huggingface.co/NousResearch/Hermes-3-Llama-3.1-8B
[16] https://huggingface.co/TsinghuaC3I/Llama-3.1-8B-UltraMedical
[17] https://huggingface.co/shenzhi-wang/Llama3.1-8B-Chinese-Chat
[18] https://huggingface.co/datasets/NousResearch/hermes-function-calling-v1
[19] https://huggingface.co/datasets/camel-ai/biology

- Chinese: We use *allenai/WildChat-1M*[20] dataset, and filter for turn=1 and language = Chinese.

We reserve 1000 prompts in each dataset for the test set, and sample subsets of the remaining prompts as domain training data.

For Function Calling domain, we investigate data accessibility scenario II. We use *argilla/apigen-function-calling*[21] dataset (APIGen-FC) as domain dataset D', which exhibits some domain shift from Hermes-FC dataset but both belong to Function Calling domain.

B.2 Training Hyperparameters

	Batch Size	# Epochs	Learning Rate
Online	8	1	1e-6
Offline	8	3	2e-5
SFT	8	3	2e-5

Table 6: Training hyperparameters for different methods, if not otherwise specified for ablations.

C Additional Details on Tables & Figures

This section provides more details on some tables and figures in the main text.

C.1 Table 1

For each line in Table 1, we use a small generic model as the draft model and compare its acceptance rates when decoding a general target model and a domain target model of the same size. The choice of models and evaluation domain datasets are specified in 7. We randomly select 1000 prompts from each dataset as the evaluation set.

	Generic Draft Model	Generic Target Model	Domain Target Model	Domain Data
Biology	Llama-3.1-8B[22]	Llama-3.1-70B[23]	OpenBioLLM-70B[24]	CAMEL-Bio[25]
Chinese	Llama-3.1-8B[26]	Llama-3.1-70B[27]	Llama3.1-70B-ZH[28]	WildChat[29] Chinese split
Coding	Qwen2.5-1.5B[30]	Qwen2.5-7B[31]	Qwen2.5-Coder-7B[32]	BigCodeBench[33]
Math	Qwen2.5-7B[34]	Qwen2.5-72B[35]	Qwen2.5-Math-72B[36]	Hendrycks-Math[37]

Table 7: Models and datasets selection.

[20] https://huggingface.co/datasets/allenai/WildChat-1M
[21] https://huggingface.co/datasets/argilla/apigen-function-calling
[22] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
[23] https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct
[24] https://huggingface.co/aaditya/Llama3-OpenBioLLM-70B
[25] https://huggingface.co/datasets/camel-ai/biology
[26] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
[27] https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct
[28] https://huggingface.co/shenzhi-wang/Llama3.1-70B-Chinese-Chat
[29] https://huggingface.co/datasets/allenai/WildChat
[30] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
[31] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[32] https://huggingface.co/Qwen/Qwen2.5-Coder-7B-Instruct
[33] https://huggingface.co/datasets/bigcode/bigcodebench
[34] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[35] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[36] https://huggingface.co/Qwen/Qwen2.5-Math-72B-Instruct
[37] https://huggingface.co/datasets/EleutherAI/hendrycks_math

C.2 FIGURE 2

Figure 2 comprehensively compare the results from different training methods. For Biology and Chinese, the training sample size is 19k; for Function Calling, the training sample size is 4k. Online distillation uses LR=1e-6; offline distillation and SFT use LR=2e-5. For offline KL and SFT, evaluation is done after three-epoch training.

D ABLATIONS

D.1 OPTIMAL LRS FOR ONLINE VS. OFFLINE DISTILLATION

We notice that online distillation requires a smaller learning rate for stable training. Across all domains, LR=1e-6 yields better performance than LR=2e-5 (see Table 8). In contrast, offline distillation exhibits the opposite trend, benefiting from a higher learning rate. This suggests that online training is more sensitive to learning rate selection, requiring careful tuning to avoid instability.

Domain	**Method**	**LR=1e-6**	**LR=2e-5**
Function Calling	Online KL	**68.7 (+13.9%)**	65.4 (+8.5%)
	Offline KL	76.4 (+26.6%)	**82.6 (+37.0%)**
Biology	Online KL	**35.4 (+5.4%)**	33.6 (+0.1%)
	Offline KL	37.9 (+12.6%)	**39.4 (+17.2%)**
Chinese	Online KL	**30.2 (+2.5%)**	27.1 (-8.0%)
	Offline KL	31.0 (+5.0%)	**33.6 (+14.0%)**

Table 8: Average acceptance rate (%) and relative change from baseline for online and offline distillation across different domains.

D.2 COMPARISON OF FORWARD KL AND REVERSE KL

When evaluating different divergence measures, we notice that forward KL and reverse KL perform comparably at lower learning rates (1e-6). However, at a higher learning rate (2e-5) which yields better results for both losses, forward KL consistently outperforms reverse KL across different training sample sizes and domains (as shown in Figure 2 and Figure 3), making it the preferred choice for offline distillation.