

Adapting While Learning: Grounding LLMs for Scientific Problems with Intelligent Tool Usage Adaptation

Bohan Lyu^{*1} Yadi Cao^{*2} Duncan Watson-Parris² Leon Bergen² Taylor Berg-Kirkpatrick² Rose Yu²

Abstract

Large Language Models (LLMs) demonstrate promising capabilities in solving simple scientific problems but often produce hallucinations for complex ones. While integrating LLMs with tools can increase reliability, this approach typically results in over-reliance on tools, diminishing the model’s ability to solve simple problems through basic reasoning. In contrast, human experts first assess problem complexity using domain knowledge before choosing an appropriate solution approach. Inspired by this human problem-solving process, we propose a novel two-component fine-tuning method. In the first component **World Knowledge Distillation (WKD)**, LLMs learn directly from solutions generated using tool’s information to internalize domain knowledge. In the second component **Tool Usage Adaptation (TUA)**, we partition problems into easy and hard categories based on the model’s direct answering accuracy. While maintaining the same alignment target for easy problems as in WKD, we train the model to intelligently switch to tool usage for more challenging problems. We validate our method on six scientific benchmark datasets, spanning mathematics, climate science and epidemiology. On average, our models demonstrate a 28.18% improvement in answer accuracy and a 13.89% increase in tool usage precision across all datasets, surpassing state-of-the-art models including GPT-4o and Claude-3.5.

1. Introduction

Large Language Models (LLMs) (Brown, 2020; Dubey et al., 2024) have demonstrated impressive capabilities across various scientific domains, from answering general

questions (Lu et al., 2022; Zhang et al., 2024b) to contributing to scientific discoveries (Ma et al., 2024; Kumar et al., 2023; Liu et al.). However, scholars note that their abilities are capped at approximately high-school levels (Rein et al., 2024; Cobbe et al., 2021; Hendrycks et al.).

Compared to direct reasoning alone, leveraging scientific tools such as physics-based simulators enables the handling of more complex problem-solving tasks. Following this trajectory, integrating LLMs with such specialized tools represents a natural evolution to enhance their problem-solving capabilities (Schick et al., 2023; Qin et al., 2024; Liu et al.). However, when trained solely on tool usage, LLMs tend to over-rely on tools, even for problems solvable through basic reasoning. This not only increases computational costs due to resource-intensive scientific tools but also limits the model’s ability to internalize knowledge.

Drawing inspiration from human expert behavior (Payne et al., 1993; Stevenson et al., 1986; Kruger & Dunning, 1999), we observe that experts first assess a scientific problem’s complexity before deciding whether to employ basic reasoning or specialized tools. We aim to instill similar adaptive capabilities in LLMs, developing them into reliable and accurate assistants across scientific domains. While previous works have explored prompt engineering and post-processing techniques (Li et al., 2024; Wan et al., 2024; Wang et al., 2024a; Zheng et al., 2024a) to improve inference accuracy and efficiency, none have focused on training models to make adaptive decisions about tool usage.

To this end, we propose a novel training paradigm consisting of two components. The first component, **World Knowledge Distillation (WKD)**, uses supervised fine-tuning and preference learning to align a pre-trained LLM with highly accurate solutions generated using information from external tools, aiming to internalize scientific knowledge. In the second component, **Tool Usage Adaptation (TUA)**, we evaluate the LLM’s direct answering ability and classify questions as easy or hard based on the model’s accuracy. While maintaining the same alignment target for easy questions, we train the model to follow external tool traces for hard questions, enabling intelligent switching based on problem complexity.

^{*}Equal contribution ¹Tsinghua University. Work partially done during Bohan Lyu’s visit to UC San Diego. ²University of California, San Diego. Correspondence to: Rose Yu <roseyu@ucsd.edu>.

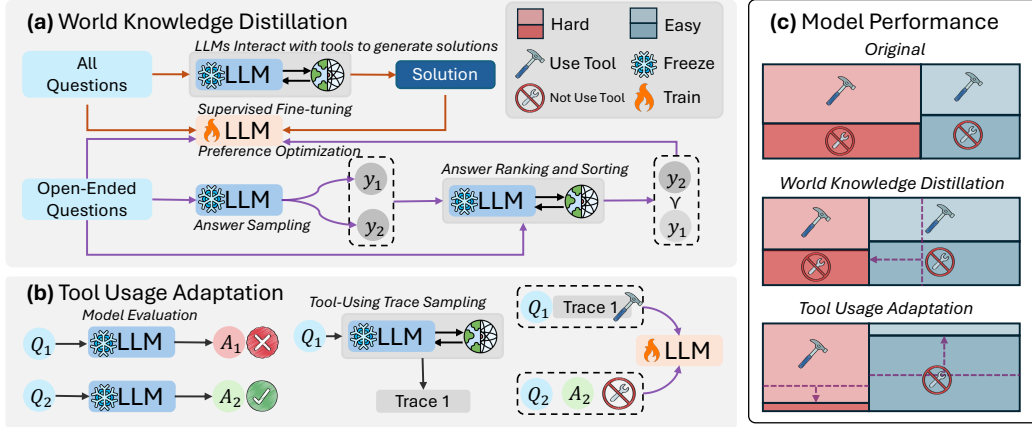


Figure 1. Pipeline of Our Method: (a) World Knowledge Distillation: LLMs undergo supervised fine-tuning (and preference learning for open-ended questions, more details in Section 5.6); (b) Tool Usage Adaptation: LLMs learn tool-usage traces for harder questions based on easy/hard question partitioning; (c) Model improvement visualization: Blue and red represent easy and hard questions, respectively. Leftward movement of the vertical dashed line indicates more questions can be solved internally; Movements of horizontal lines for easy/hard questions respectively show more intelligent tool usage decisions.

For empirical evaluation, we incorporated datasets across diverse scientific domains, from classic textbook-level math and physics problems to research frontiers like climate science and epidemiology. The experimental results show significant improvements after applying our method, especially on our custom-created datasets, which focus on challenging and specialized questions that pretrained LLMs had not encountered.

Our contributions are summarized as follows:

- We introduce a novel two-stage training paradigm that enables LLMs to adaptively solve real-world scientific problems of varying complexity.
- We construct four additional datasets spanning various scientific domains, including both questions and solutions, to facilitate future research in this direction.
- Experiments on both custom and public datasets demonstrate the effectiveness of our work, resulting in better answer accuracy and more intelligent tool use.

2. Related Work

LLM Alignment Alignment techniques aim to make LLMs behave in accordance with human values, using methods such as supervised fine-tuning (SFT) (Zhang et al., 2024a) and reinforcement learning (RL). RL methods can be categorized into online (Rafailov et al., 2024; Meng et al., 2024) and offline (Schulman et al., 2017; Ouyang et al., 2022; Lee et al., 2023; Bai et al., 2022) algorithms. Both online and offline algorithms convert the non-trivial scoring of text outputs into pairwise preference orderings.

We employ SFT for all questions in our dataset. Additionally, we utilize RL to learn preferences between different proposals for open-ended questions.

Grounding LLMs for Scientific Problems Previous work has sought to ground LLMs using domain-specific knowledge across various scientific fields: climate science (Thulke et al., 2024), biomedical science (Luo et al., 2022), molecular science (Chithrananda et al., 2020), and general science (Zhang et al., 2024b; Taylor et al., 2022). Most of these approaches heavily rely on expert annotations or distillation from stronger models and face scalability limitations due to computational and expert labor costs.

These constraints highlight the need to integrate scientific tools into both data generation and training processes.

LLM as Tool User Researchers have integrated LLMs with various tools (Schick et al., 2023; Tang et al., 2023; Patil et al., 2023; Qin et al., 2023b; Wang et al., 2024c), which fall into two broad categories: 1. tools with time-varying results, such as search engines (Nakano et al., 2022; Qin et al., 2023a) and social media platforms (Park et al., 2023; Ye et al., 2023); 2. tools with consistent results, such as compilers, physics-based simulators (Ma et al., 2024; Kumar et al., 2023; Liu et al.; Bran et al., 2023; Huang et al., 2024), and scientific knowledge bases (Kraus et al., 2023; Koldunov & Jung, 2024; Thulke et al., 2024; Vaghefi et al., 2023). While these approaches leverage LLMs’ tool-using capabilities, they do not enhance the LLMs’ inherent domain knowledge. Furthermore, existing studies have not addressed training LLMs to make intelligent decisions about tool usage based on problem complexity, often resulting in over-reliance on the tools covered during training.

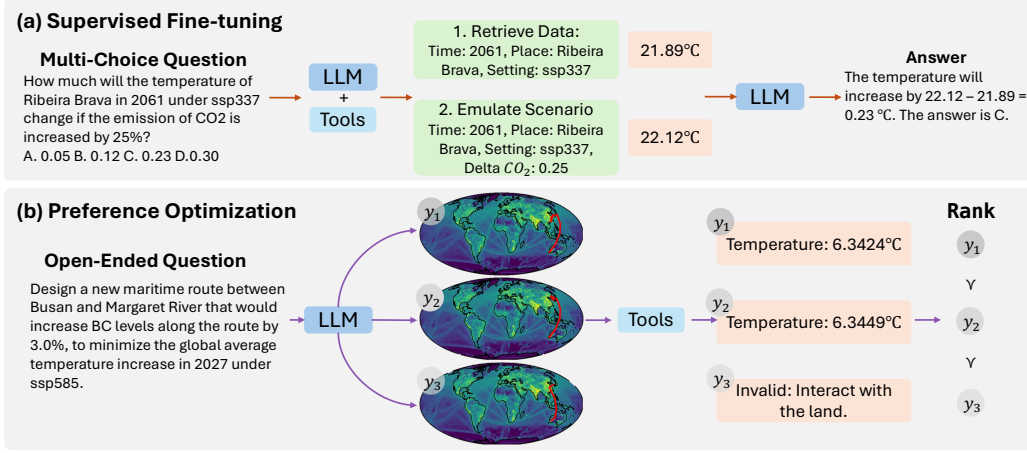


Figure 2. WKD approaches for different question types: (a) for all problems, we prompt the LLMs to use tools to generate deterministic solutions, then use those solutions as targets for supervised fine-tuning (SFT). (b) When extended to open-ended questions (detailed in Section 5.6), apart from the first step, LLM generates an ensemble of proposals, which are ranked using pre-defined metrics to construct preference pairs; preference optimization is then applied using these data.

These limitations highlight the need for a training approach that enables LLMs to use tools adaptively, striking a balance between their inherent knowledge and external tool utilization.

3. Methodology

To equip LLMs with both internalized knowledge and adaptive tool usage capabilities, we propose a two-component training paradigm: World Knowledge Distillation (WKD) and Tool Usage Adaptation (TUA). The first component equips the model with domain-specific expertise through supervised fine-tuning and preference optimization. The second component trains the model to intelligently leverage external tools based on task complexity. Together, these components enable the model to balance direct reasoning with effective tool use, enhanced accuracy and reliability in scientific problem-solving. Figure 1 illustrates our fine-tuning pipeline and its intended target.

3.1. Solution Generation with Tools

Our approach integrates professional tools E , such as physics simulators, with LLMs for high-accuracy answer generation. We employ templates for both questions and corresponding tool traces t for solution generation. At each step in t , we instruct the LLM to run the simulator using a specific format, i.e., a system prompt P_f instructing and forcing tool usage. After gathering the returned information $\{I_e\}_t$ from tool usage traces t , the LLM generates the solution y using its policy π combined with the context provided in question x . We denote this whole process as:

$$y \sim \pi(\cdot \mid x, \{I_e\}_t, P_f). \quad (1)$$

Figure 2 demonstrates the solution generation pipeline and its application to both multiple-choice and open-ended questions.

3.2. World Knowledge Distillation (WKD)

Following the generation of these solutions, we proceed to directly fine-tune the target LLM. The alignment loss between the generated answer and (1) is defined as:

$$J_{\text{Direct}}(\theta, \mathcal{D}, P) = -\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(\cdot \mid x, \{I_e\}_t, P_f)} [\log \pi_\theta(y \mid x, P)], \quad (2)$$

where \mathcal{D} represents the training dataset. The loss for World Knowledge Distillation (WKD) is then:

$$J_{\text{WKD}}(\theta, \mathcal{D}) = J_{\text{Direct}}(\theta, \mathcal{D}, P_n), \quad (3)$$

where P_n is the prompt that allows no tool usage.

The aim of WKD is to prompt the LLM to generate solutions directly, without relying on tools. This is the ideal case if achievable. However, when problems are too challenging and hallucination may occur, we switch to Tool Usage Adaptation (TUA), which trains the LLM to intelligently select tools to use based on problem complexity.

3.3. Tool Usage Adaptation (TUA)

The TUA begins by evaluating the fine-tuned LLMs after WKD on the questions of a benchmark. For each question, we sample an ensemble of directly generated answers to calculate the accuracy rate. Based on a predefined threshold of the accuracy rate, we partition the questions into two subsets: $\mathcal{D}_{\text{easy}}$, problems the LLM can solve directly, and $\mathcal{D}_{\text{hard}}$, the remaining ones.

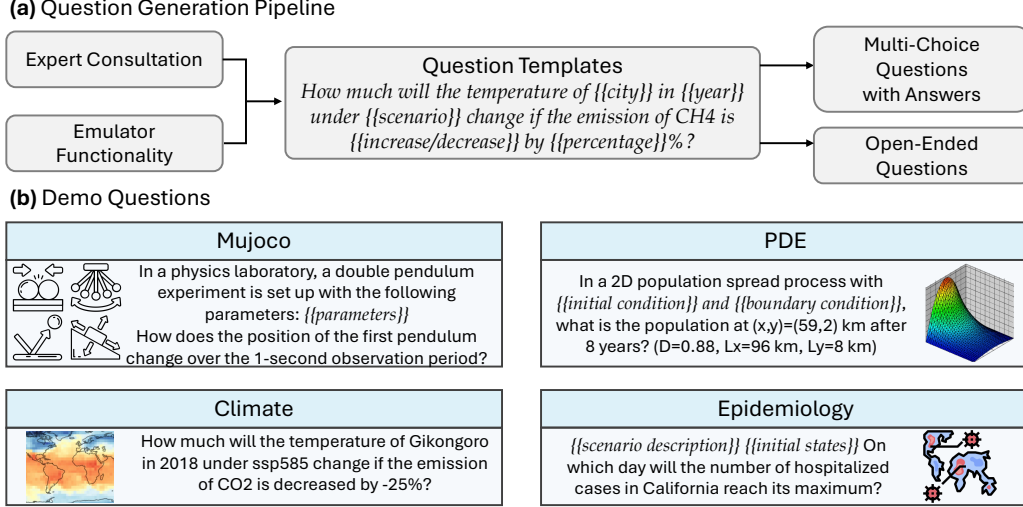


Figure 3. (a) Question generation pipeline using templates and (b) Selected demo questions from our custom-created datasets.

For $\mathcal{D}_{\text{easy}}$, we keep the alignment target as in (2). However, for $\mathcal{D}_{\text{hard}}$, we switch the alignment target to the augmented solution with tool usage traces and train the LLM to follow these traces accurately. In this case, the alignment loss for correct traces is:

$$J_{\text{Trace}}(\theta, \mathcal{D}, P) = -\mathbb{E}_{x \sim \mathcal{D}, t \sim \pi(\cdot | x, E, P_f)} \log \pi_{\theta}(t | x, E, P). \quad (4)$$

The combined training loss considering both easy and hard questions is:

$$J_{\text{TUA}}(\theta, \mathcal{D}_{\text{easy}}, \mathcal{D}_{\text{hard}}) = \lambda J_{\text{Direct}}(\theta, \mathcal{D}_{\text{easy}}, P_i) + (1 - \lambda) J_{\text{Trace}}(\theta, \mathcal{D}_{\text{hard}}, P_i), \quad (5)$$

where P_i is a prompt that allows LLMs to intelligently choose whether to utilize external tools or not. λ adjusts the weight between the two subsets to prevent extreme proportion distribution. In our case, we set $\lambda = 0.5$ as our dataset’s easy and hard questions have proportions at the same magnitude. Note that we apply the same prompt P_i across both easy and hard questions to train the model’s ability to determine when to leverage tools.

3.4. Knowledge Consistency Across Prompt Strategies

In our settings, certain knowledge required for directly answering questions should be learned under both prompting strategies: P_n during WKD, and P_i during TUA and during deployment. Recent research (Zeng et al., 2024) has highlighted a critical challenge: knowledge acquired under one prompting strategy may not readily transfer to another, often resulting in significant performance degradation. We encountered similar issues in our preliminary experiments when attempting to alternate between WKD and TUA within training iterations.

To mitigate this issue, we propose leveraging a mixed loss that simultaneously considers both WKD and TUA objectives, thereby maintaining consistent knowledge across different prompting strategies.

For each epoch, we first partition the dataset into easy and hard questions following the procedure outlined in Section 3.3. The mixed loss function is defined as:

$$J_{\text{Mix}}(\theta, \mathcal{D}, \mathcal{D}_{\text{easy}}, \mathcal{D}_{\text{hard}}) = \alpha J_{\text{WKD}}(\theta, \mathcal{D}) + (1 - \alpha) J_{\text{TUA}}(\theta, \mathcal{D}_{\text{easy}}, \mathcal{D}_{\text{hard}}), \quad (6)$$

where α is a hyperparameter that balances the learning under P_n and P_i . In our experiments, we consistently set $\alpha = 0.5$. We note that this term differs fundamentally from a re-weighting of the loss terms in Equation 5, as the latter only takes the input as P_i .

4. Experiments

4.1. Dataset

We employ two existing public datasets, MATH and SciBench, and construct four new scientific datasets for our experiments: Mujoco, Partial Differential Equations (PDEs), Climate Science, and Epidemiology. Detailed statistics of all datasets are available in Appendix A.1.

Our custom dataset construction follows a systematic pipeline. First, we design domain-specific question templates based on both the expert consultation and the emulator functionality. We then generate individual questions by sampling parameters within scientifically valid ranges. Finally, we produce corresponding solutions using LLMs and tool usage traces, as detailed in Section 3.2.

Our custom datasets primarily comprise multiple-choice questions (MCQs), with the Climate dataset additionally

Table 1. Answer Accuracy (%) across different datasets and models. All baselines use prompt P_n (no tool usage). Our baseline model are evaluated with P_n and P_f (forced tool usage), respectively. Model after AWL training is evaluated with both P_n (internal knowledge) and P_i (intelligent tool usage), respectively. We highlight results ranked as **first** and **second**.

Models	Mujoco	PDE	Climate	Epidemiology	MATH	SciBench	Average
Llama3.1-70B	46.79	55.83	37.50	30.83	73.73	45.00	48.28
GPT4o	52.86	69.17	35.83	32.50	81.92	71.67	57.32
GPT4o-mini	51.79	70.83	30.00	35.83	<u>80.79</u>	<u>68.33</u>	56.26
Claude3.5-Sonnet	48.57	65.83	32.50	35.00	80.23	67.50	54.94
Llama3.1-8B (Base)- P_n	28.57	31.09	30.83	21.67	54.24	17.50	30.65
Llama3.1-8B (Base)- P_f	<u>59.32</u>	61.67	77.50	<u>57.78</u>	69.23	31.67	<u>59.53</u>
Llama3.1-8B-Ours- P_n	55.00	<u>75.00</u>	<u>80.00</u>	51.11	61.02	30.83	58.83
Llama3.1-8B-Ours- P_i	64.47	78.33	83.33	74.44	62.15	34.17	66.15

incorporating open-ended questions (such as policy proposals for climate change mitigation). Open-ended questions, lacking definitive “gold answers”, necessitate a modified pipeline for preference learning—details of which will be presented in Section 5.6. The publicly available MATH and SciBench datasets consist exclusively of numerical problems.

MATH MATH (Hendrycks et al.) is a collection of high-school-level competition mathematics problems, categorized into five difficulty levels. Following previous work (Qian et al., 2023), we utilize problems from the test set with numerical answers to evaluate our methods.

SciBench SciBench (Wang et al., 2024b) is a challenging collegiate-level benchmark of scientific problems covering Mathematics, Physics, and Chemistry.

Mujoco We developed this dataset to address problems related to rigid- and soft-body dynamics, utilizing the Mujoco physics engine (Todorov et al., 2012). Unlike idealized scenarios common in textbooks, these problems incorporate real-world complexities such as stiffness, damping, and various types of collisions and frictional forces.

Partial Differential Equations (PDEs) We created in-house numerical solvers and design questions in heat transfer, chemical engineering, population simulation, etc. These questions require solving 1D or 2D PDEs.

Climate Science We designed a dataset of climate science problems centered around a neural surrogate model (Niu et al., 2024). The model accepts inputs including time, climate scenario (e.g., ssp126, ssp245), and emissions of greenhouse gases (CO_2 , CH_4) and aerosol gases (BC , SO_2), outputting the corresponding earth surface temperature.

Epidemiology We constructed this dataset of epidemiological problems using a state-of-the-art surrogate model (Wu et al., 2023). The model inputs include 28-day, multi-dimensional features for each California county,

along with a 24-dimensional state-level initial conditions describing the epidemiology state. It outputs the predicted epidemiology state over the subsequent 28 days.

Example questions are illustrated in Figure 3. Detailed dataset statistics and question templates are provided in Appendix A.2, while comprehensive demonstration questions can be found in Appendix A.3.

4.2. Experiment Setup

Models We used Llama-3.1-8B-Instruct (Dubey et al., 2024) as the base model for our training scheme. We also conducted extensive evaluations of other state-of-the-art (SOTA) open and closed-source models, including GPT4o, GPT4o-mini, Claude-3.5-Sonnet, and Llama-3.1-70B-Instruct.

Comparisons We primarily report the performance of the untrained base model both with and without tool usage, as well as its performance after training with our method. Additionally, we conducted ablation studies to analyze the influence of various factors, such as the necessity of WKD and TUA and the proportion of noisy data in the training dataset.

Experiments with P_n were conducted under a zero-shot setting for our custom-created datasets and MATH, and under a few-shot setting for SciBench with officially provided shots. For experiments with P_f and P_i , we sampled correct tool traces generated by GPT-4o as few-shot prompts.

Tools We employed different tools for each dataset. For Mujoco, we used the official APIs. In the case of PDEs, we utilized in-house numerical solvers. For Climate and Epidemiology datasets, we employed APIs wrapping the corresponding neural surrogate models. For open-ended datasets, we used a Python code interpreter.

4.3. Evaluation Metrics

We primarily evaluate two types of accuracy: Answer Accuracy and Tool Usage Accuracy.

Table 2. The Accuracy of Tool Usage. The models after AWL demonstrate remarkable accuracy across all datasets. In contrast, most other models show accuracy around 50% which indicates an inability to make intelligent decisions on tool usage.

Models	Mujoco	PDE	Climate	Epidemiology	MATH	SciBench	Average
Llama3.1-70B	49.66	50.00	48.67	48.94	55.77	50.93	50.66
GPT4o	50.30	52.41	48.70	50.57	49.54	50.00	50.25
GPT4o-mini	50.34	52.35	48.81	61.84	55.19	68.36	56.15
Claude3.5-Sonnet	50.39	51.27	49.38	54.95	51.57	54.37	51.99
Llama3.1-8B (Base)	51.50	50.00	50.35	50.86	49.52	60.22	52.07
Llama3.1-8B-Ours	61.80	66.67	75.50	66.61	62.46	62.75	65.96

Answer Accuracy Answer accuracy quantifies the proportion of correct answers provided by the models. For multiple-choice questions (MCQs) in our custom-created datasets, we assign binary scores based on whether the model selects the correct choice. For numerical answers in the MATH and SciBench datasets, we consider answers correct if they fall within a small tolerance range of the true value, specifically within $\pm 5\%$.

Tool Usage Accuracy Tool usage accuracy assesses whether the model can make intelligent decisions regarding tool usage, i.e., using tools for difficult questions while answering directly for easier ones. Questions are partitioned into Easy (E) or Hard (H) based on whether they are answerable by a trained model with P_n (no tool usage). When using P_i , which allows tool choice, the decisions are further labeled with T (tool used) or N (no tool used). For example, HT indicates a hard question where the model chose to use a tool. Our tool usage accuracy is defined as $\frac{1}{2} \times (\frac{EN}{EN+ET} + \frac{HT}{HN+HT})$.

5. Results

5.1. Answer Accuracy

We report the answer accuracy comparison across all datasets in Table 1. Our method substantially outperforms all baselines on custom-created datasets that were not commonly covered in pre-training. While our model does not surpass current state-of-the-art models on public datasets, it shows significant improvement over the base model without fine-tuning. This performance gap on public benchmarks is likely due to the larger parameter count and specific optimization of state-of-the-art models on the open-source datasets.

5.2. Tool Usage Accuracy

We present the tool usage accuracy in Table 2. Overall, our trained model achieves the best tool usage accuracy across all datasets, except for SciBench where it ranks second, demonstrating the ability to make intelligent decisions on tool usage. In contrast, other models exhibit accuracy around 50% = $\frac{1}{2}(0 + 100\%)$, indicating two typical cases:

either over-reliance on tools or never attempting to use them (empirical proof is in Appendix C.3).

Besides the advantage shown in Table 2, we further investigate the tool usage decisions on the MATH dataset, which provides a prior label of difficulty levels in Figure 8. Our trained model exhibits a reasonable increase in tool usage with growing question difficulty, while the base model shows an over-reliance on tools regardless of difficulty. In contrast, Claude-3.5 demonstrates more confidence in answering directly for both easy and hard questions, possibly because MATH is a public dataset and the model has seen similar questions during training.

Miscellaneous Results on Tool Usage For the sake of conciseness in the main text, we include additional miscellaneous results on tool usage in Appendix C. Specifically, Appendix C.1 provides additional metrics for analysis; Appendix C.2 shows the evolution of tool usage decisions over training epochs; and Appendix C.3 compares the tool usage decisions of our method and Claude-3.5 on our custom-created and open datasets, respectively.

5.3. Functionality of Sub-components

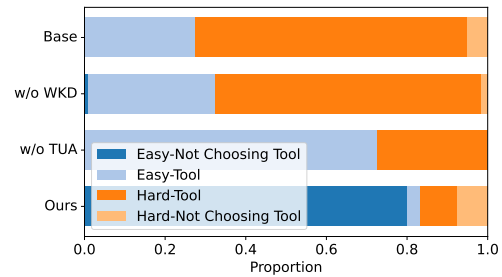


Figure 4. Composition of Tool Usage Decisions in Climate Dataset Training: Impact of individual training components in ablation study.

Figure 4 presents an ablation study on the functionality of WKD and TUA by evaluating the proportion of the four tool usage decisions (EN, ET, HN, HT) on the Climate dataset. We observe that omitting either component leads to tool over-reliance. Moreover, without WKD, the model exhibits the lowest answer accuracy, as it is never trained on the distilled knowledge directly.

This ablation shows the necessity of both components in our approach: WKD for knowledge internalization and TUA for intelligently switching to tool usage.

5.4. Functionality of Prompt Strategies

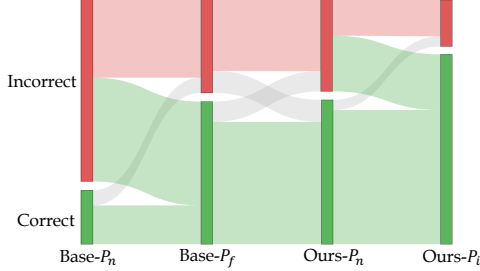


Figure 5. Different models’ performance on the PDE Dataset: comparing pre- and post-training, with and without tool usage.

We present the model’s performance on the PDE dataset in Figure 5 before and after training, and with P_n and P_i , respectively.

The model Ours- P_n demonstrates performance comparable to Base- P_f , both showing a significant improvement over the base model. This similarity indicates successful internalization of distilled knowledge from tools. The transition from Ours- P_n to Ours- P_i showcases further improvement in answer accuracy, resulting from the model’s enhanced ability to intelligently switch to tools for harder questions.

5.5. Resistance to Noisy Data brought by TUA

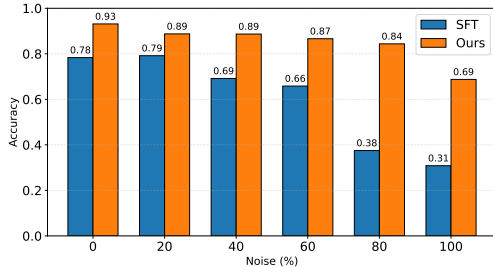


Figure 6. Model Performance vs. Noise Level: Comparison between our two-component method and SFT-only approaches on Climate dataset.

Generating solutions via LLMs or human expert annotation inevitably introduces noise. We examine how model trained with our method performs with increasing noise on the Climate dataset. We compare the model trained with our two-component method, against a model that only undergoes SFT for comparison. The results are shown in Figure 6.

The WKD-only model’s performance degrades drastically with increasing levels of noise, as the underlying distribution becomes polluted. However, this does not significantly impact the trained model with P_i . The model judges these

polluted questions as harder and opts to use tools to ensure accuracy, demonstrating the robustness of our approach. As noise levels increase, the performance of the SFT-only method declines, while models trained with our method demonstrate robust performance.

5.6. Extension to Open-ended Questions

We extend our approach to open-ended questions, specifically addressing problems with two key characteristics: 1. Expensive verification, reflecting the high cost of experimental or simulation as tools. This constraint precludes the use of model-based reinforcement learning, which would otherwise be a mature solution. 2. High penalties for violating feasibility constraints, mirroring real-world applications where failed plans incur unaffordable costs (e.g., aircraft design). Such questions necessitate modifications to both data generation and the training pipeline to incorporate preference learning with our method:

Modified Data Generation We prompt the LLM to generate an ensemble of solutions for a given problem (e.g., proposing a marine route between two ports that minimizes future temperature rise). These solutions are evaluated using professional tools (e.g., a neural surrogate model for climate science) to derive pre-defined metrics L , such as a penalty function of temperature rise. We then form pair-wise rankings for subsequent preference learning. We expand the definition of trace t' to include: 1) ensemble generation, 2) ensemble evaluation with tools, and 3) ranking.

Enhanced Training Pipeline

- **WKD:** We switch to preference alignment using DPO based on pair-wise rankings of proposals. For infeasible proposals, we assign $-\infty$ to their L to effectively

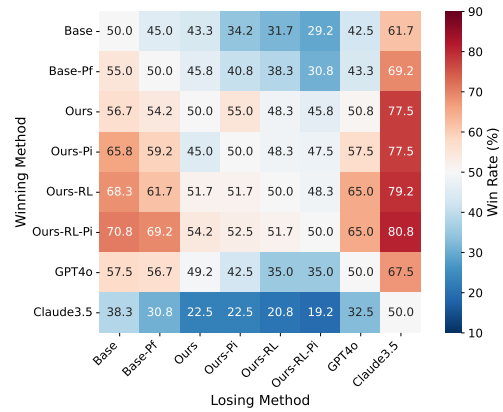


Figure 7. Win rate heatmap of the percentage that each model won in pairwise comparisons against other models. Each cell represents the win rate (%) of the model listed on the y-axis when compared with the model on the x-axis.

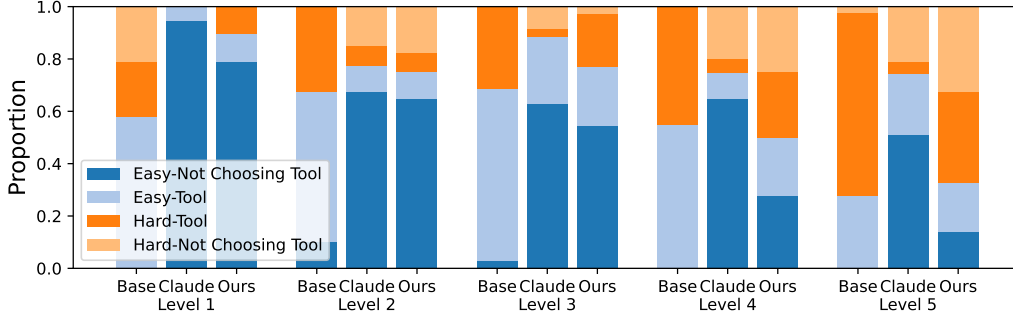


Figure 8. Tool usage decision of different models on MATH dataset of 5 difficulty levels. Investigated models are **Claude-3.5-Sonnet**, **Llama-3.1-8B-Base**, and **Llama-3.1-8B-Ours**.

filter them out from the ensembles.

- **Easy/Hard Problem Determination:** We gauge problem difficulty by setting a quality threshold and measuring the frequency of generated feasible and qualified proposals rather than answer accuracy rates.
- **TUA:** For hard problems, the LLM adopts a more cautious approach, generating and evaluating multiple proposals with tools (within a budget cap) before selecting the best. This process naturally uses the expanded trace definition t' , in contrast to MCQs where a single tool call suffices.

We denote the model trained with our original approach as “Ours” and the one with preference learning as “Ours-RL”.

Table 3. Percentage of qualified responses that satisfy the constraints and meet a pre-established quality threshold.

Base	Base- P_f	Ours	Ours- P_i
31.82	29.09	37.50	40.17
Ours-RL	Ours-RL- P_i	GPT4o	Claude3.5
47.50	49.16	34.17	31.51

We evaluate these models alongside several baselines on climate-related open-ended questions. Table 3 reports the percentage of qualified proposals, and Figure 7 shows win-rate comparisons among models. The results show improvements from incorporating both preference learning and the TUA. The contribution of these components can be attributed to: (1) preference learning implicitly learning the ranking among diverse proposals for each problem, and (2) the TUA enabling intelligent tool switching for sampling and ranking proposals on harder questions.

Table 4 shows the tool usage accuracy across models. Compared to both the base model and closed-source alternatives, our trained model achieves better discrimination in when tools are necessary while reducing the overall frequency of tool usage.

Table 4. Tool Usage Accuracy (\uparrow , first line) and Tool Usage Rate (\downarrow , second line) across different models, respectively.

Tool Usage Metrics	GPT4o	Claude3.5	Base	Ours- P_i
Accuracy (%) \uparrow	50.00	50.00	49.37	56.57
Rate (%) \downarrow	92.50	100.00	100.00	55.82

6. Conclusion and Future Works

We introduced a novel two-component fine-tuning approach to enhance Large Language Models (LLMs) in solving scientific problems of varying complexity. Our approach equips LLMs with the ability to intelligently choose between using appropriate tools or conducting basic reasoning independently by assessing problem difficulty using their internalized knowledge, resembling human expert problem-solving strategies. Experiments across diverse datasets demonstrate that our fine-tuning method significantly improves a smaller base model’s performance, enabling it to surpass larger models such as GPT-4o and Claude-3.5. On average, our fine-tuned models achieve a 28.18% increase in answer accuracy and a 13.89% improvement in tool usage precision across all datasets.

We expect our method to serve as a paradigm and foundation for creating reliable AI scientific assistants, and we note several promising directions for future investigation: Our current approach requires domain-specific fine-tuning, future research could explore methods for unifying cross-domain training in related scientific fields. Incorporating step-wise adaptive tool utilization, i.e., adaptive decision-making on tool usage at each step, could significantly reduce human preprocessing workload. Finally, expanding our method to handle multi-modal inputs and outputs would broaden its applicability to settings where data extends beyond textual formats.

7. Acknowledgment

This work was supported in part by the U.S. Army Research Office under Army-ECASE award W911NF-07-R-0003-03,

the U.S. Department Of Energy, Office of Science, IARPA HAYSTAC Program, and NSF Grants #2205093, #2146343, #2134274, CDC-RFA-FT-23-0069, DARPA AIE FoundSci and DARPA YFA.

References

- Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.
- Brown, T. B. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- 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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Huang, K., Qu, Y., Cousins, H., Johnson, W. A., Yin, D., Shah, M., Zhou, D., Altman, R., Wang, M., and Cong, L. Crispr-gpt: An llm agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.
- Koldunov, N. and Jung, T. Local climate services for all, courtesy of large language models. *Communications Earth & Environment*, 5(1):13, 2024.
- Kraus, M., Bingler, J., Leippold, M., Schimanski, T., Senni, C. C., Stambach, D., Vaghefi, S., and Webersinke, N. Enhancing large language models with climate resources. Technical report, Swiss Finance Institute, 2023.
- Kruger, J. and Dunning, D. Unskilled and unaware of it: how difficulties in recognizing one’s own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999.
- Kumar, V. V., Gleyzer, L., Kahana, A., Shukla, K., and Karniadakis, G. E. Mycrunchgpt: A chatgpt assisted framework for scientific machine learning. *Journal of Machine Learning for Modeling and Computing*, 2023.
- Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Hall, E., Carbune, V., Rastogi, A., et al. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- Li, Y., Yuan, P., Feng, S., Pan, B., Wang, X., Sun, B., Wang, H., and Li, K. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. 2024.
- Liu, R., Wei, J., Gu, S. S., Wu, T., Vosoughi, S., Cui, C., Zhou, D., and Dai, A. M. Mind’s eye: Grounded language model reasoning through simulation. In *The Eleventh International Conference on Learning Representations*.
- Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K., Zhu, S., Tafford, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., and Liu, T. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409, 2022.
- Ma, P., Wang, T., Guo, M., Sun, Z., Tenenbaum, J. B., Rus, D., Gan, C., and Matusik, W. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *International Conference on Machine Learning*. PMLR, 2024.
- Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback, 2022.
- Niu, R., Wu, D., Kim, K., Ma, Y., Watson-Parris, D., and Yu, R. Multi-fidelity residual neural processes for scalable surrogate modeling. In *International Conference on Machine Learning, ICML 2024, Proceedings of Machine Learning Research*, 2024.

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Park, J. S., O’Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.
- Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis, 2023.
- Payne, J. W., Bettman, J. R., and Johnson, E. J. *The Adaptive Decision Maker*. Cambridge University Press, New York, 1993. doi: 10.1017/CBO9781139173933.
- Qian, C., Han, C., Fung, Y., Qin, Y., Liu, Z., and Ji, H. CREATOR: Tool creation for disentangling abstract and concrete reasoning of large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6922–6939, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.462.
- Qin, Y., Cai, Z., Jin, D., Yan, L., Liang, S., Zhu, K., Lin, Y., Han, X., Ding, N., Wang, H., Xie, R., Qi, F., Liu, Z., Sun, M., and Zhou, J. Webcpm: Interactive web search for chinese long-form question answering. In *Proceedings of ACL 2023*. Association for Computational Linguistics, 2023a.
- Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023b.
- Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Huang, Y., Xiao, C., Han, C., Fung, Y. R., Su, Y., Wang, H., Qian, C., Tian, R., Zhu, K., Liang, S., Shen, X., Xu, B., Zhang, Z., Ye, Y., Li, B., Tang, Z., Yi, J., Zhu, Y., Dai, Z., Yan, L., Cong, X., Lu, Y., Zhao, W., Huang, Y., Yan, J., Han, X., Sun, X., Li, D., Phang, J., Yang, C., Wu, T., Ji, H., Liu, Z., and Sun, M. Tool learning with foundation models, 2024.
- 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, 2024.
- Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.
- Stevenson, H. W., Azuma, H. E., and Hakuta, K. E. Child development and education in japan. In *This book is based on a conference sponsored by the Center for Advanced Study in the Behavioral Sciences; the Japan Society for the Promotion of Science; and the Joint Committee on Japanese Studies of the American Council of Learned Societies and the Social Science Research Council*. WH Freeman/Times Books/Henry Holt & Co, 1986.
- Tang, Q., Deng, Z., Lin, H., Han, X., Liang, Q., and Sun, L. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases, 2023.
- Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.
- Thulke, D., Gao, Y., Pelsner, P., Brune, R., Jalota, R., Fok, F., Ramos, M., Wyk, I., Nasir, A., Goldstein, H., et al. Climategpt: Towards ai synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*, 2024.
- Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.
- Vaghefi, S., Wang, Q., Muccione, V., Ni, J., Kraus, M., Binger, J., Schimanski, T., Senni, C. C., Webersinke, N., Huggel, C., and Leippold, M. Chatclimate: Grounding conversational ai in climate science. *Swiss Finance Institute Research Paper No. 23-88*, April 11 2023. doi: 10.2139/ssrn.4414628.
- Wan, G., Wu, Y., Chen, J., and Li, S. Dynamic self-consistency: Leveraging reasoning paths for efficient llm sampling. *arXiv preprint arXiv:2408.17017*, 2024.
- Wang, X., Feng, S., Li, Y., Yuan, P., Zhang, Y., Pan, B., Wang, H., Hu, Y., and Li, K. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. *arXiv preprint arXiv:2408.13457*, 2024a.

- Wang, X., Hu, Z., Lu, P., Zhu, Y., Zhang, J., Subramaniam, S., Loomba, A. R., Zhang, S., Sun, Y., and Wang, W. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. In *Proceedings of the Forty-First International Conference on Machine Learning*, 2024b.
- Wang, Z., Cheng, Z., Zhu, H., Fried, D., and Neubig, G. What are tools anyway? a survey from the language model perspective. *arXiv preprint arXiv:2403.15452*, 2024c.
- Wu, D., Niu, R., Chinazzi, M., Vespignani, A., Ma, Y., and Yu, R. Deep bayesian active learning for accelerating stochastic simulation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2023.
- Ye, Y., Cong, X., Tian, S., Cao, J., Wang, H., Qin, Y., Lu, Y., Yu, H., Wang, H., Lin, Y., Liu, Z., and Sun, M. Proagent: From robotic process automation to agentic process automation, 2023.
- Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y., and Tang, J. AgentTuning: Enabling generalized agent abilities for LLMs. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 3053–3077, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.181.
- Zhang, B., Liu, Z., Cherry, C., and Firat, O. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024a.
- Zhang, D., Hu, Z., Zhoubian, S., Du, Z., Yang, K., Wang, Z., Yue, Y., Dong, Y., and Tang, J. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning, 2024b.
- Zheng, Y., Li, P., Yan, M., Zhang, J., Huang, F., and Liu, Y. Budget-constrained tool learning with planning. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 9039–9052, Bangkok, Thailand and virtual meeting, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.536.
- Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024b. Association for Computational Linguistics.

A. Dataset Details

We utilize two existing public datasets, MATH and SciBench, alongside four custom scientific datasets that we developed: Mujoco, Partial Differential Equations (PDEs), Climate Science, and Epidemiology. Below, we provide detailed descriptions of the datasets, along with the tools employed to construct and evaluate them.

A.1. Dataset Statistics

Table 5 shows the statistics of the seven datasets used in our experiments. For our custom datasets (Mujoco, PDE, Climate, and Epidemiology), we show the number of scenarios and question templates used to generate the problems. The existing datasets (MATH and SciBench) are from established benchmarks that do not provide information about scenarios and templates.

Table 5. Statistics of the datasets: number of questions in training and test sets, and number of scenarios and question templates where applicable. MATH and SciBench are from existing benchmarks that do not provide information about scenarios and templates.

Dataset	Train Questions	Test Questions	Scenario	Templates
Mujoco	960	280	9	53
PDE	1627	120	36	5
Climate	640	120	5	19
Epidemiology	1720	90	1	4
MATH	1600	171	-	-
SciBench	266	120	-	-
Open-Ended	582	120	1	1

A.2. Dataset Components

A.2.1. MUJOCO

We developed the Mujoco dataset to address problems in rigid- and soft-body dynamics. This dataset is based on the Mujoco physics engine (Todorov et al., 2012), which simulates realistic physics scenarios. Previous work introduced a dataset comprising 39 qualitative questions and trained LLMs to solve them using MuJoCo simulations. However, this benchmark has proven to be too simplistic for current models, which can achieve 100% accuracy with ease. To address this limitation, we have developed a new dataset consisting of 8 distinct scenarios of different complexity based on a public tutorial¹. Each scenario contains an average of 14.5 adjustable parameters, including variables such as the initial position and velocity of objects, time constants, damping ratios, friction coefficients, and the gravitational acceleration of the environment.

A.2.2. PDE (PARTIAL DIFFERENTIAL EQUATION)

The PDE dataset focuses on solving partial differential equations in fields such as heat transfer, chemical engineering, and population dynamics. We wrote 1-D and 2-D partial differential equation solvers for the diffusion process, which can be set with different variables like diffusion coefficient and size of the field, and different kinds of initial situations and boundary situations with different parameters.

A.2.3. CLIMATE

The Climate Science dataset comprises problems related to predicting earth surface temperature changes based on climate scenarios. The dataset is built using a neural surrogate model (Niu et al., 2024) that integrates data across multiple fidelity levels for robust climate modeling. The model utilizes 12 climate driver variables as input, encompassing total emissions of greenhouse gases (CO₂, CH₄) and the first five principal components of global aerosol gas (BC, SO₂) distributions, derived from a 72x96 global grid. The output predicts air temperature 2 meters above the Earth’s surface at a global scale. The model spans historical data from 1850-2015 and projects future scenarios from 2015 to 2100 under four Shared Socioeconomic Pathways (SSPs): ssp126, ssp245, ssp370, and ssp585. These scenarios range from sustainable development with low challenges to mitigation and adaptation (ssp126) to fossil-fueled development with high challenges to mitigation

¹<https://pab47.github.io/mujocopy.html>

and adaptation (ssp585), representing a spectrum of potential future climate states and associated societal responses.

A.2.4. EPIDEMIOLOGY

The Epidemiology dataset focuses on simulating disease spread and predicting epidemiological states over time. This dataset is based on a state-of-the-art surrogate model (Wu et al., 2023) that predicts disease progression using multi-dimensional input features. For the California epidemic scenario, the input consists of two components: 1. county-level data for 58 counties, including 24 features per county per day over 28 days, 2. 10 initial state-level features. The output predicts 10 state-level features for each of the next 28 days.

A.2.5. MATH

MATH (Hendrycks et al.) is a widely used benchmark that consists of high-school-level mathematics competition problems. The dataset covers various topics such as algebra, geometry, and number theory, and is divided into five difficulty levels. It remains challenging compared with another renowned math dataset GSM8K (Cobbe et al., 2021), where current 7B LLMs already achieve over 80% accuracy. Following previous work (Qian et al., 2023), we utilize problems from the MATH test set with definite numerical answers to evaluate our methods.

A.2.6. SCIBENCH

SciBench (Wang et al., 2024b) is a collegiate-level benchmark that includes scientific problems in fields such as Mathematics, Physics, and Chemistry. Like MATH, the problems are numerical and focus on real-world scientific applications. We use the SciBench dataset to evaluate models on complex numerical problems.

A.3. Question Examples

We provide question examples in our custom-created datasets with different scenarios and question templates.

<p>In a physics laboratory, a double pendulum experiment is set up with the following parameters:</p> <ul style="list-style-type: none"> - Gravitational acceleration: -9.61 m/s^2 - Mass of first pendulum rod: 0.1 kg - Mass of first pendulum bob: 0.07 kg - Mass of second pendulum rod: 0.17 kg - Mass of second pendulum bob: 0.2 kg - Sliding friction coefficient: 0.11 - Torsional friction coefficient: 0.68 - Rolling friction coefficient: 0.21 - Initial angle of the first pendulum: 0.98 radians - Initial angular velocity of the first pendulum: 0.86 rad/s - Initial angle of the second pendulum: 2.21 radians - Initial angular velocity of the second pendulum: -0.87 rad/s <p>The pendulum is released and its motion is observed for 5 seconds.</p> <p>How does the position of the second pendulum change over the 5-second observation period?</p> <p>(A) Stable (B) Steady increase by 14.4% (C) Fluctuating, decrease by 40.3% (D) Fluctuating, overall stable</p>
<p>In a physics laboratory, a rolling ball experiment is set up with the following parameters:</p> <ul style="list-style-type: none"> - Gravitational acceleration: 9.27 m/s^2 - Initial position: 0.79 meters - Radius of the ball: 0.12 meters - Mass of the ball: 2.78 kg - Sliding friction coefficient: 0.58 - Torsional friction coefficient: 0.35 - Rolling friction coefficient: 0.2 - Initial velocity: -1.15 m/s (X), 4.01 m/s (Z) - Initial angular velocity: 1.27 rad/s (Y) - Damping coefficient: 0.23 <p>The ball is rolled and its motion is observed for 1 seconds.</p> <p>What is the range of X positions (in meters) that the ball occupies during its motion in the 1-second observation period?</p> <p>(A) $[-0.36, -0.27]$ (B) $[-0.27, -0.18]$ (C) $[-0.18, -0.09]$ (D) $[-0.09, -0.00]$</p>

Figure 9. Example questions in the Mujoco Dataset.

<p>What is the average temperature of Palenga in 1869?</p> <p>(A) 21.084519958496 (B) 23.720084953308 (C) 26.355649948120 (D) 28.991214942932</p>
<p>What is the temperature of Toumoukro in 2035 under ssp370 if the emission of CH4 is increased by 25%?</p> <p>(A) 22.5064071655273 (B) 25.3197080612183 (C) 28.1330089569092 (D) 30.9463098526001</p>
<p>How much will the temperature of Al Hamalah in 2047 under ssp126 change if the emission of CH4 is decreased by -10%?</p> <p>(A) -0.02068302 (B) -0.00741459 (C) -0.01771736 (D) -0.02278250</p>
<p>What is the range of temperature of Soweto in 2063 under different climate settings?</p> <p>(A) [21.80831527709961, 22.57936096191406] (B) [22.57936096191406, 23.35040664672852] (C) [23.35040664672852, 24.12145233154297] (D) [24.12145233154297, 24.89249801635742]</p>
<p>For Ebreichsdorf, Gleisdorf, Perchtoldsdorf, Voitsberg, which city has the lowest temperature in 2058 under ssp245?</p> <p>(A) Perchtoldsdorf (B) Ebreichsdorf (C) Gleisdorf (D) Voitsberg</p>
<p>What is the minimum level of agreement we should support if we want to control the temperature of Rocha in 2083 under 18.706981430053713?</p> <p>(A) ssp370 (B) ssp245 (C) ssp585 (D) ssp126</p>

Figure 10. Example questions (MCQs) in the Climate Dataset.

<p>Design a new maritime route between Singapore (lon: 103.8, lat: 1.3) and Dubai (lon: 55.2972, lat: 25.2631) that would increase SO2 levels along the route by 3.0%. Propose a route that would minimize the global average temperature increase in 2033 under ssp126. Present your answer as a list of coordinates (longitude, latitude) representing key points along the route. Format your response as follows: [(longitude_1, latitude_1), (longitude_2, latitude_2), ..., (longitude_n, latitude_n)]. Include at least the starting point, endpoint, and any significant waypoints. Ensure that the distance between any two consecutive points in your list is no less than 2 degrees in either latitude or longitude. Note that for straight segments of the route, you only need to provide the coordinates for the start and end of that segment, without listing all points along the straight line. The route will be automatically connected based on the nodes you provide.</p>

Figure 11. Example open-ended question in the Climate Dataset.

In a 1D chemical diffusion experiment, the initial concentration is uniformly set at 28 mol/L. Dirichlet boundary conditions are applied, with the concentration fixed at 13 mol/L at $x = 0$ and 6 mol/L at $x = L$, where $L = 4$ cm. The diffusion coefficient is $D = 0.0007 \text{ cm}^2/\text{s}$. After 253 seconds, what is the maximum concentration (mol/L)?

- (A) 19.502
- (B) 22.288
- (C) 25.074
- (D) 27.86**

In a 1D population spread process, the initial population density is 60 individuals/km² for $x < L/2$ and 30 individuals/km² for $x \geq L/2$, with Neumann boundary conditions (zero flux at the boundaries). The domain length is $L = 44$ km and the diffusion coefficient is $D = 0.68 \text{ km}^2/\text{year}$. What is the maximum population density (individuals/km²) after 9 years?

- (A) 60.0**
- (B) 66.0
- (C) 72.0
- (D) 78.0

In a 2D heat conduction experiment, the initial temperature follows a checkerboard pattern with alternating regions of 100 °C and 0 °C. Dirichlet boundary conditions are applied with temperatures of 8 °C, 14 °C, 73 °C, and 21 °C at the left, right, bottom, and top boundaries, respectively. The domain dimensions are $L_x = 65$ cm and $L_y = 6$ cm, and the diffusion coefficient is $D = 0.21 \text{ cm}^2/\text{s}$. After 356 seconds, what is the minimum temperature (°C)?

- (A) 8.0**
- (B) 8.9
- (C) 7.1
- (D) 10.4

In a 2D chemical diffusion experiment, the initial concentration follows a checkerboard pattern with alternating regions of 100 mol/L and 0 mol/L. Neumann boundary conditions (zero flux at the boundaries) are used, with the domain dimensions set to $L_x = 1$ cm and $L_y = 10$ cm. The diffusion coefficient is $D = 0.0006 \text{ cm}^2/\text{s}$. After 1000 seconds, what is the maximum concentration (mol/L)?

- (A) [-3.5049231554707703, 20.00626361945248)
- (B) [20.00626361945248, 37.74154059285945)
- (C) [37.74154059285945, 82.61383728899432)**
- (D) [82.61383728899432, 97.32889694911078)

In a 1D chemical diffusion experiment, the initial concentration is set at 75 mol/L. Dirichlet boundary conditions are applied, with the concentration fixed at 88 mol/L at $x = 0$ and 4 mol/L at $x = L$, where $L = 4$ cm. The diffusion coefficient is $D = 0.0009 \text{ cm}^2/\text{s}$. After 50 seconds, what is the maximum gradient of concentration (mol/L per cm)?

- (A) 144.82**
- (B) 159.302
- (C) 173.784
- (D) 188.266

Figure 12. Example questions in the PDE Dataset.

In an epidemiological study simulating the spread of disease across California, daily data from 58 counties over 28 days is used to model disease transmission dynamics. Each county has 10 input features per day, and the model is initialized with 24 state-level features.

The 10 county-level features are:

- 0: seasonality min
- 1: omega community interventions
- 2: omega work interventions
- 3: omega school interventions
- 4: omega home interventions
- 5: alpha school interventions
- 6: transit commute interventions
- 7: international travel interventions
- 8: domestic travel interventions
- 9: R0

The 24 state-level features are:

- 0: prevalence CA state total Latent
- 1: prevalence CA state total Infectious symptomatic
- 2: prevalence CA state total Infectious asymptomatic
- 3: prevalence CA state total Hospitalized
- 4: prevalence CA state total ICU
- 5: prevalence CA state total Removed asymptomatic
- 6: prevalence CA state total Removed symptomatic
- 7: prevalence CA state total Home asymptomatic
- 8: prevalence CA state total Home mild
- 9: prevalence CA state total Home severe
- 10: prevalence CA state total Removed hospitalized
- 11: prevalence CA state total Deaths hospitalized
- 12: incidence CA state total Latent
- 13: incidence CA state total Infectious symptomatic
- 14: incidence CA state total Infectious asymptomatic
- 15: incidence CA state total Hospitalized
- 16: incidence CA state total ICU
- 17: incidence CA state total Removed asymptomatic
- 18: incidence CA state total Removed symptomatic
- 19: incidence CA state total Home asymptomatic
- 20: incidence CA state total Home mild
- 21: incidence CA state total Home severe
- 22: incidence CA state total Removed hospitalized
- 23: incidence CA state total Deaths hospitalized

In this simulation, the following data is provided:

{{simulation parameters}}

Given the data, on which day does the maximum value of incidence CA state total ICU occur?

- (A) Day 7
- (B) Day 1
- (C) Day 25**
- (D) Day 9

Figure 13. Example question in the Epidemiology Dataset.

A.4. Prompt Examples

P_n for our custom-created datasets:

```
Answer the following question. Your answer should be in the following format:
Solution: <Your solution process>
Answer: <Your answer, one of A/B/C/D>

Question: {{question}}
```

P_i for our custom-created datasets:

```
Given the following functions, please respond with a JSON for a function call with its
proper arguments that best answers the given prompt.

Respond in the format {"name": function name, "parameters": dictionary of argument name
and its value}. Do not use variables.

{{functions}}

If you don't know the answer, you can use the tool to help you. If you can answer the
problem without the tool, answer the problem directly.

Question: {{question}}
```

P_n for SciBench and MATH:

```
Answer the following question. Your answer should be in the following format:
Solution: <Your solution process>
Answer: <Your answer, a pure number>

Question: {{question}}
```

P_i for SciBench and MATH:

```
Please answer the following question. You can write code to solve the problem or give the
answer directly. When answering, you should first give the Solution then give the Answer
. The answer should be a pure number without LaTeX or unit signs. Each time, you should
either write code or answer the question. Your final answer should be in one of the
following formats:

If you want to write code, your answer should be in the following format:

Thought: <Your thought>
Action: write_and_run_code
Code:
```python
<Your code>
```

If you want to answer the question, you should answer in the following format:

Thought: <Your thought>
Action: answer_question
Solution: <Your solution>
Answer: <Your answer>

Question: {{question}}
```


For P_f , we remove descriptions about intelligent tool usage in the above P_i prompts, requiring the use of tools. If the model directly answers the question, we will ask the model to use tools before answering.

A.5. Solution Examples

A.5.1. EXAMPLE FROM THE CLIMATE DATASET

Problem:

Question: How much will the temperature of Ewo in 2068 under ssp126 change if the emission of CO2 is decreased by -25%?

(A) -0.13081711
(B) -0.16986465
(C) -0.09639389
(D) -0.05745936

Correct Answer: B

Problem-solving without tool usage:

To answer this question, I would need to know the specifics of the climate model used (e.g., CMIP6 model) and the scenario used (ssp126). However, since this information is not provided, I'll provide a general framework for how one might approach this problem.

1. Choose a climate model: Select a suitable climate model that has been run under the ssp126 scenario.
2. Select a location: Identify the location of Ewo (presumably a small town in the Democratic Republic of Congo).
3. Run the climate model: Run the climate model for the year 2068 under the ssp126 scenario with a -25% reduction in CO2 emissions.
4. Extract the temperature change: Extract the temperature change at the location of Ewo for the year 2068 from the climate model output.
5. Compare with options: Compare the extracted temperature change with the provided options to select the correct answer.

However, without access to the climate model output or the specific details of the scenario, I'll provide a placeholder answer.

Solution: Run the climate model for the year 2068 under the ssp126 scenario with a -25% reduction in CO2 emissions and extract the temperature change at the location of Ewo.

Therefore, the answer is D.

Problem-solving with tool usage.

Assistant: Call query_lat_and_lon with parameters {'city_name': 'Ewo'}

Feedback from query_lat_and_lon: latitude: -0.8742, longitude: 14.8167.

Assistant: Call diy_greenhouse with parameters {'longitude': 14.8167, 'latitude': -0.8742, 'setting': 'ssp126', 'year': 2068, 'delta_CO2': -0.25, 'delta_CH4': 0}

Feedback from diy_greenhouse: The temperature is 27.52265167236328.

Assistant: Call future_temperature with parameters {'longitude': 14.8167, 'latitude': -0.8742, 'year': 2068, 'setting': 'ssp126'}

Feedback from future_temperature: The temperature is 27.692516326904297.

Assistant: Call answer_question with parameters {'thought': 'The temperature for Ewo in 2068 under the ssp126 scenario with no changes in emissions is 27.692516326904297. With a 25% decrease in CO2 emissions, the temperature is 27.52265167236328. The difference is 27.692516326904297 - 27.52265167236328 = 0.16986465454101562.', 'answer': 'B', 'solving_process': '1. Retrieved the latitude and longitude of Ewo. 2. Predicted the temperature for 2068 under the ssp126 scenario with a 25% decrease in CO2 emissions. 3. Retrieved the temperature for 2068 under the ssp126 scenario with no changes in emissions. 4. Calculated the difference between the two temperatures to find the change.'}

Solution generated based on the information from tool usage, which is then used for World Knowledge Distillation training:

To calculate the temperature change in Ewo by 2068 under the ssp126 scenario with a 25% decrease in CO₂ emissions, we first identify Ewo's location at latitude -0.8742 and longitude 14.8167. The temperature for Ewo in 2068 under the ssp126 scenario with no changes in emissions is determined to be 27.6925°C. With a 25% decrease in CO₂ emissions, the temperature is calculated to be 27.5227°C.

The difference between the two temperatures is:

$$27.6925^{\circ}\text{C} - 27.5227^{\circ}\text{C} = 0.1699^{\circ}\text{C}$$

This matches option B from the given choices. Therefore, the answer is **B**.

A.5.2. EXAMPLE FROM THE SCIBENCH

Problem:

Question: A 10.0 g block with a charge of $+8.00 \times 10^{-5}\text{C}$ is placed in an electric field $\vec{E} = (3000\hat{i} - 600\hat{j})\text{N/C}$. What is the magnitude of the electrostatic force on the block? The unit of the answer is N.

Correct Answer: 0.245

Problem-solving without tool usage:

Solution: The force \vec{F} on the block is the product of the electric field \vec{E} and the charge q of the block:

$$\vec{F} = q\vec{E}.$$

Calculations: We can find F with the equation for the magnitude of a vector and the given values of E and q :

$$\begin{aligned} F &= |q\vec{E}| = |q|E = \sqrt{E_x^2 + E_y^2} \\ &= \sqrt{(3000\text{ N/C})^2 + (-600\text{ N/C})^2} \\ &= \sqrt{9.00 \times 10^6\text{ N}^2/\text{C}^2 + 3.60 \times 10^5\text{ N}^2/\text{C}^2} \\ &= \sqrt{9.36 \times 10^6\text{ N}^2/\text{C}^2} = \sqrt{9.36 \times 10^{12}\text{ N}^2/\text{C}^2} \\ &= 3.06 \times 10^6\text{ N/C}. \end{aligned}$$

Answer: 3060000 (×)

Problem-solving with tool usage:

Assistant:

Thought: To find the magnitude of the electrostatic force on the block, we will use the formula $F = qE$, where q is the charge of the block and E is the magnitude of the electric field. Since the electric field is a vector, we first need to find its magnitude using the Pythagorean theorem.

Action: write_and_run_code

Code:

```
# Given data
q = 8.00e-5 # charge in C
E_x = 3000 # electric field in N/C
E_y = -600 # electric field in N/C

# Magnitude of the electric field
E = (E_x**2 + E_y**2)**0.5
print(f"Magnitude of the electric field (E): {E} N/C")

# Electrostatic force
F = q * E
print(f"Electrostatic force (F): {F} N")
```

Tool Feedback: Magnitude of the electric field (E): 3059.411708155671 N/C Electrostatic force (F): 0.2447529366524537 N

Assistant:

Thought: The magnitude of the electric field was calculated using the Pythagorean theorem, and the electrostatic force was then found by multiplying the charge of the block by the magnitude of the electric field. The calculated electrostatic force is 0.2447529366524537 N. Action: answer_question Solution:

First, find the magnitude of the electric field E using the Pythagorean theorem:

$$E = \sqrt{E_x^2 + E_y^2}$$

Given: - $E_x = 3000 \text{ N/C}$ - $E_y = -600 \text{ N/C}$

Substitute the values:

$$E = \sqrt{(3000 \text{ N/C})^2 + (-600 \text{ N/C})^2}$$

$$E = \sqrt{9000000 + 360000}$$

$$E = \sqrt{9360000}$$

$$E = 3059.411708155671 \text{ N/C}$$

Next, find the electrostatic force F using the formula:

$$F = qE$$

Given: - Charge of the block $q = 8.00 \times 10^{-5} \text{ C}$

Substitute the values:

$$F = (8.00 \times 10^{-5} \text{ C}) \times (3059.411708155671 \text{ N/C})$$

$$F = 0.2447529366524537 \text{ N}$$

Therefore, the magnitude of the electrostatic force on the block is 0.2447529366524537 N.

Answer: 0.244753 (✓)

Solution generated based on the information from tool usage, which is then used for World Knowledge Distillation training:

Solution: We aim to determine the magnitude of the electrostatic force on a 10.0 g block with a charge of $+8.00 \times 10^{-5} \text{ C}$ placed in an electric field $\vec{E} = (3000\hat{i} - 600\hat{j}) \text{ N/C}$.

Given Data:

- **Charge of the Block (q):** $+8.00 \times 10^{-5} \text{ C}$
- **Electric Field (\vec{E}):** $(3000\hat{i} - 600\hat{j}) \text{ N/C}$

Analysis:

1. Magnitude of the Electric Field (E):

- To find the magnitude of the electric field, we use the Pythagorean theorem:

$$E = \sqrt{E_x^2 + E_y^2}$$

- **Given Values:**

$$E_x = 3000 \text{ N/C}$$

$$E_y = -600 \text{ N/C}$$

- **Substituting Values:**

$$E = \sqrt{(3000 \text{ N/C})^2 + (-600 \text{ N/C})^2}$$

$$E = \sqrt{9000000 + 360000}$$

$$E = \sqrt{9360000}$$

$$E = 3059.411708155671 \text{ N/C}$$

2. Electrostatic Force (F):

- The electrostatic force on the block is given by the formula:

$$F = qE$$

- **Given Values:**

$$q = 8.00 \times 10^{-5} \text{ C}$$

- **Substituting Values:**

$$F = (8.00 \times 10^{-5} \text{ C}) \times (3059.411708155671 \text{ N/C})$$

$$F = 0.2447529366524537 \text{ N}$$

Conclusion:

The magnitude of the electrostatic force on the block is 0.244753 N.

Answer: 0.244753 N

B. Fintune Backbone Details

For training, we employ Llama-Factory (Zheng et al., 2024b) as the LLM training platform. Table 6 shows our training hyperparameters for both supervised fine-tuning and DPO training. For the preference optimization training in Section 5.6, we first conduct supervised fine-tuning using the preferred answers from the preference dataset, then apply LoRA for DPO training to ensure model robustness. All training is performed on an L40S server.

For inference, we utilize the API of proprietary models. Specifically, we access Llama-3.1-70B-Instruct through the API provided by <https://groq.com>.

Table 6. Hyperparameters for supervised fine-tuning and DPO training with LoRA.

| <i>Full-parameter Supervised Fine-tuning</i> | |
|--|--------|
| Parameter | Value |
| Train batch size | 64 |
| Learning rate | 1.0e-5 |
| Number of epochs | 3.0 |
| LR scheduler | cosine |
| Warmup ratio | 0.1 |
| Precision | bf16 |
| <i>DPO Training with LoRA</i> | |
| Parameter | Value |
| LoRA target | all |
| LoRA rank | 8 |
| DPO beta | 0.1 |
| Train batch size | 32 |
| Learning rate | 5.0e-6 |
| Number of epochs | 3.0 |
| LR scheduler | cosine |
| Warmup ratio | 0.1 |
| Precision | bf16 |

C. Additional Analysis of Tool Usage Accuracy

C.1. Other Metrics for Analysis

Here we provide a detailed analysis of tool usage accuracy across various models and datasets. We first elucidate the categorization of tool usage decisions in Table 7. In the table, we categorize decisions into four types based on problem difficulty (Easy or Hard) and tool usage choice (Tool or Not Choosing Tool). Easy problems are those that the model can answer correctly without using tools, while Hard problems are those that the model cannot answer correctly without assistance. The Tool or Not Choosing Tool distinction represents the model’s decision to use or not use tools when given the option. Therefore, EN (Easy problems solved without tools) and HT (Hard problems solved with tools) are expected in the aspect of intelligent tool usage.

Table 7. Explanation of Tool Usage Decision, where \checkmark indicates the expected decisions: not choosing tools for easy problems (EN) and using tools for hard problems (HT).

| | Tool (T) | Not Choosing Tool (N) |
|----------|---------------------|-----------------------|
| Easy (E) | ET | EN (\checkmark) |
| Hard (H) | HT (\checkmark) | HN |

The following tables present various metrics to evaluate tool usage across different models and datasets. Table 8 employs a balanced measure of tool usage accuracy, computed as $\frac{1}{2} \times (\frac{EN}{EN+ET} + \frac{HT}{HN+HT})$, giving equal weight to performance on both problem types to address potential dataset imbalances. Tables 9 and 10 disaggregate this metric into easy and hard problem categories, measured by $\frac{EN}{EN+ET}$ and $\frac{HT}{HT+HN}$ respectively. These assess the models’ ability to recognize when tool usage is unnecessary for simpler tasks and beneficial for complex problems. Table 11 presents the raw accuracy of tool usage decisions without accounting for potential class imbalances, computed as $\frac{EN+HT}{EN+ET+HT+HN}$. Finally, Table 12 quantifies the proportion of total tool usage, calculated as $\frac{ET+HT}{EN+ET+HT+HN}$, with lower values indicating more selective tool use.

Table 8. The Accuracy of Tool Usage, measured with $\frac{1}{2} \times (\frac{EN}{EN+ET} + \frac{HT}{HN+HT})$.

| Models | Mujoco | PDE | Climate | Epidemiology | MATH | SciBench | Average |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Llama3.1-70B | 49.66 | 50.00 | 48.67 | 48.94 | <u>55.77</u> | 50.93 | 50.66 |
| GPT4o | 50.30 | <u>52.41</u> | 48.70 | 50.57 | 49.54 | 50.00 | 50.25 |
| GPT4o-mini | 50.34 | 52.35 | 48.81 | <u>61.84</u> | 55.19 | 68.36 | <u>56.15</u> |
| Claude3.5-Sonnet | 50.39 | 51.27 | 49.38 | 54.95 | 51.57 | 54.37 | 51.99 |
| Llama3.1-8B | <u>51.50</u> | 50.00 | <u>50.35</u> | 50.86 | 49.52 | 60.22 | 52.07 |
| Llama3.1-8B-Ours | 61.80 | 66.67 | 75.50 | 66.61 | 62.46 | <u>62.75</u> | 65.96 |

Table 9. The Accuracy of Tool Usage for easy problems, measured with $\frac{EN}{EN+ET}$.

| Models | Mujoco | PDE | Climate | Epidemiology | MATH | SciBench | Average |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Llama3.1-70B | 0.00 | 0.00 | 0.00 | 2.70 | 95.40 | 85.19 | 30.55 |
| GPT4o | 1.35 | <u>4.82</u> | 0.00 | 30.77 | 83.45 | 0.00 | 20.06 |
| GPT4o-mini | 0.69 | 4.71 | 0.00 | <u>41.86</u> | <u>83.92</u> | 68.29 | <u>33.24</u> |
| Claude3.5-Sonnet | 1.47 | 2.53 | 0.00 | 38.10 | 80.28 | <u>72.84</u> | 32.54 |
| Llama3.1-8B | <u>5.00</u> | 0.00 | <u>2.08</u> | 3.85 | 5.21 | 44.00 | 10.02 |
| Llama3.1-8B-Ours | 47.40 | 86.67 | 96.00 | 52.08 | 71.30 | 35.14 | 64.76 |

C.2. The Evolution of Tool Usage Accuracy with Training Epochs

Figure 14 illustrates the evolution of our model’s performance in the form of different solution types (EN, ET, HN, HT) on the Climate dataset at different training epochs.

As training progresses, we observe a significant increase in the proportion of correct direct answers (blue bars), indicating successful knowledge internalization. Additionally, there is a notable decrease in tool over-reliance (initially, orange and

Table 10. The Accuracy of Tool Usage for hard problems, measured with $\frac{HT}{HT+HN}$.

| Models | Mujoco | PDE | Climate | Epidemiology | MATH | SciBench | Average |
|------------------|---------------|---------------|--------------|--------------|--------------|---------------|--------------|
| Llama3.1-70B | <u>99.33</u> | 100.00 | 97.33 | <u>95.18</u> | 16.13 | 16.67 | 70.77 |
| GPT4o | 99.24 | 100.00 | 97.40 | 70.37 | 15.62 | 100.00 | 80.44 |
| GPT4o-mini | 100.00 | 100.00 | 97.62 | 81.82 | 26.47 | 68.42 | 79.05 |
| Claude3.5-Sonnet | 99.31 | <u>100.00</u> | 98.77 | 71.79 | 22.86 | 35.90 | 71.44 |
| Llama3.1-8B | 98.00 | 100.00 | <u>98.61</u> | 97.87 | 93.83 | 76.45 | 94.13 |
| Llama3.1-8B-Ours | 76.19 | 46.67 | 55.00 | 81.13 | <u>53.62</u> | <u>90.36</u> | 67.16 |

 Table 11. The Accuracy of Tool Usage, measured with $\frac{EN+HT}{EN+ET+HT+HN}$.

| Models | Mujoco | PDE | Climate | Epidemiology | MATH | SciBench | Average |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Llama3.1-70B | 52.86 | 44.17 | 60.83 | 66.67 | 74.58 | 47.50 | 57.77 |
| GPT4o | 47.50 | 34.17 | 62.50 | 57.50 | 71.19 | 28.33 | 50.20 |
| GPT4o-mini | 48.57 | 32.50 | <u>68.33</u> | <u>67.50</u> | <u>72.88</u> | 68.33 | 59.69 |
| Claude3.5-Sonnet | 51.79 | 35.83 | 66.67 | 60.00 | 68.93 | 60.83 | 57.34 |
| Llama3.1-8B | 71.43 | <u>68.91</u> | 60.00 | 77.50 | 45.76 | <u>71.20</u> | <u>65.80</u> |
| Llama3.1-8B-Ours | <u>60.36</u> | 76.67 | 89.17 | 67.33 | 64.41 | 73.33 | 71.88 |

 Table 12. The Proportion of Tool Usage (\downarrow), measured with $\frac{ET+HT}{EN+ET+HT+HN}$.

| Models | Mujoco | PDE | Climate | Epidemiology | MATH | SciBench | Average |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Llama3.1-70B | 99.64 | 100.00 | <u>98.33</u> | 95.83 | 7.63 | 15.83 | 69.54 |
| GPT4o | 98.93 | <u>96.67</u> | 98.33 | 70.00 | <u>16.38</u> | 100.00 | 80.05 |
| GPT4o-mini | 99.64 | 96.67 | 98.33 | 73.33 | 18.08 | 43.33 | 71.56 |
| Claude3.5-Sonnet | 98.93 | 98.33 | 99.17 | <u>68.33</u> | 20.34 | <u>30.00</u> | <u>69.18</u> |
| Llama3.1-8B | <u>97.14</u> | 100.00 | 98.33 | 97.50 | 94.35 | 73.14 | 93.41 |
| Llama3.1-8B-Ours | 63.21 | 21.67 | 12.50 | 65.35 | 38.42 | 82.50 | 47.27 |

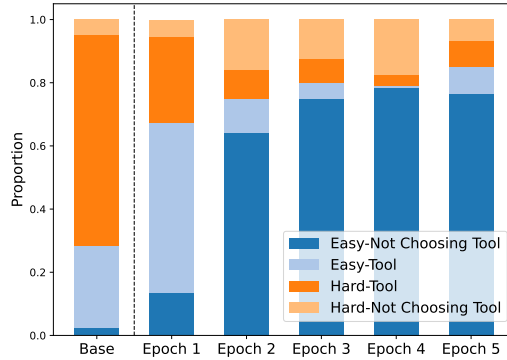
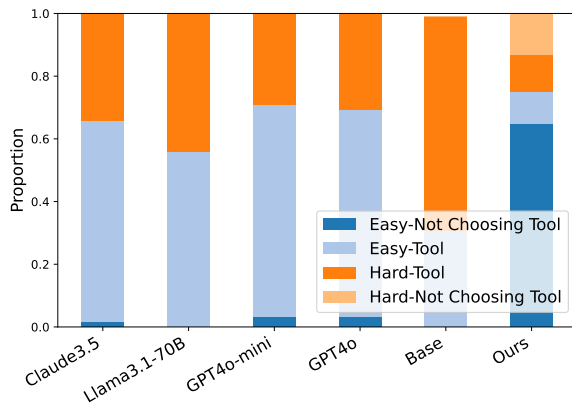


Figure 14. Composition of Tool Usage Decisions in Climate Dataset Training: Evolution over growing momentum training terms.

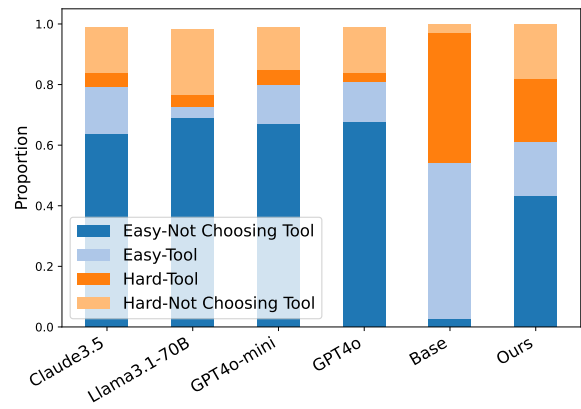
gray bars dominate nearly 100%) and an increase in tool usage for hard questions (orange bar). This demonstrates the effectiveness of our training approach in intelligently switching to tool usage only when question is hard.

C.3. Composition of Tool Usage Decisions across Open and Custom Datasets

Figure 15 illustrates the composition of tool usage decisions for different models on both custom and public datasets. We observe that for custom datasets, the closed models tend to over-rely on tools, whereas for open datasets, they tend to provide direct answers. This empirically supports our hypothesis that closed models have encountered similar questions in open datasets and are familiar with the answers.



(a) PDE Dataset



(b) MATH Dataset

Figure 15. Composition of 4 Tool Usage Decisions for Different Models on Both Custom and Public Datasets.