TaP: A Taxonomy-Guided Framework for Automated and Scalable Preference Data Generation

Anonymous ACL submission

Abstract

Conducting supervised fine-tuning and preference fine-tuning on large language models (LLMs) requires high-quality datasets to improve their ability to follow instructions and align with human preferences and values. However, constructing such datasets is resource-intensive, and most available datasets for supervised and preference fine-tuning are in English. To address these challenges, we propose the **Ta**xonomy-Guided **P**reference Data Generation (TaP) framework, which facilitates automated and scalable construction of preference datasets across various languages. TaP is grounded in a structured taxonomy that allows fine-grained control over dataset composition, thereby ensuring both diversity and comprehensive coverage. We employ TaP-generated datasets to perform supervised and preference fine-tuning on various LLMs. Experimental results demonstrate that LLMs trained on TaP-generated datasets outperform those trained on existing open-source datasets. Remarkably, LLMs trained on TaP-generated datasets surpass the performance of those trained on an open-source dataset that is 180 times larger.

1 Introduction

Large language models (LLMs) typically undergo three primary training stages: (1) pre-training, during which LLMs are trained on extensive and diverse datasets encompassing multiple languages and modalities (Zhao et al., 2023; Yang et al., 2024; Dubey et al., 2024; DeepSeek-AI et al., 2024); (2) supervised fine-tuning, in which LLMs are fine-tuned using prompt-response pairs authored by humans or LLMs (Zhang et al., 2023b); and (3) preference fine-tuning, wherein preference data annotated by humans or LLMs are employed to conduct RLHF or alternative methods such as DPO and KTO (Ouyang et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024). The latter two stages are essential for improving LLMs' ability to follow instructions and generate responses that are aligned with human preferences and values.

However, datasets for supervised and preference fine-tuning usually require human annotation, a process that is time-consuming and expensive. These constraints significantly hinder the scalability of dataset expansion, thereby limiting the volume of available training data. A common alternative involves leveraging LLMs to generate synthetic data, reducing dependence on human labor and enhancing scalability. Nevertheless, naive prompting often results in homogeneous data with limited coverage and pronounced task or domain imbalances, wherein certain domains or tasks are overrepresented while others remain underrepresented (Wang et al., 2023; Gunasekar et al., 2023; Xu et al., 2025). Such imbalances can lead LLMs to overfit to the overrepresented domains or tasks, thereby weakening the models' generalizability in real-world applications. Consequently, ensuring both diversity and comprehensive coverage in synthetic data generation remains a critical challenge.

Furthermore, compared with open-source datasets for supervised fine-tuning, there are significantly fewer datasets for preference fine-tuning (Cui et al., 2024; Wang et al., 2024d,c). Additionally, open-source datasets for both supervised and preference fine-tuning are predominantly available in English. This language bias limits the global applicability of LLMs and marginalizes non-English-speaking communities, preventing them from fully benefiting from these models. While translating English datasets into other languages presents a potential solution, it can introduce translation errors and result in "translationese", thereby compromising dataset quality (Riley et al., 2020). Given that there are over 7,000 languages worldwide, it is imperative to develop novel approaches and release open-source datasets in diverse languages to better support the latter two training stages of LLMs.

To address these challenges, we propose the

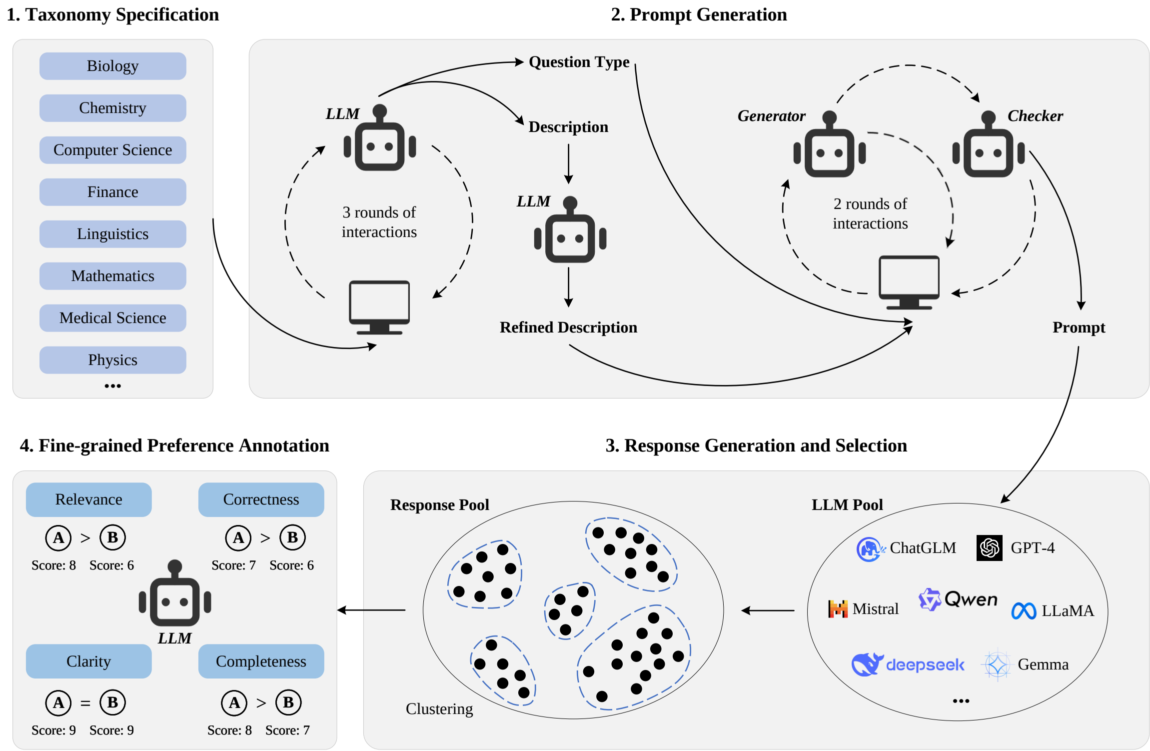

Figure 1: Diagram of the preference dataset construction process using TaP, where the taxonomy can be derived from the undergraduate program catalog.

Taxonomy-Guided **P**reference Data Generation (TaP) framework, which enables automated and scalable generation of preference data across various languages. The TaP framework leverages a predefined taxonomy to guide LLMs in producing preference data spanning a wide range of domains and tasks. By generating data in accordance with the predefined taxonomy, TaP ensures fine-grained control over dataset composition and facilitates the construction of diverse datasets with comprehensive coverage of the categories defined within the taxonomy. The TaP framework comprises four key steps: (1) designing a taxonomy aligned with the intended application of LLMs—for instance, a taxonomy for general-purpose LLMs can be organized around common use cases such as brainstorming, open-ended question answering, and summarization; (2) prompting LLMs to generate prompts corresponding to the categories specified in the taxonomy; (3) employing multiple LLMs to generate diverse responses for each prompt; and (4) utilizing LLMs to evaluate and rank these responses according to predefined evaluation criteria. A diagram of the preference dataset construction process using the TaP framework is presented in Figure 1.

Motivated by the fundamental role of knowledge in decision-making, problem-solving, and adaptation to new environments, as well as by empirical findings indicating that LLMs benefit from diverse and knowledge-rich training data (Li et al., 2023; Lozhkov et al., 2024), we propose leveraging a comprehensive taxonomy that encompasses a broad spectrum of knowledge and skills essential to human expertise to guide the generation of such preference data using LLMs. Through careful manual analysis, we identify undergraduate program catalogs as a valuable resource for constructing such a taxonomy. These catalogs serve as structured and comprehensive repositories of essential knowledge required for human learning. Considering that undergraduate education constitutes a foundational component of higher education systems worldwide, and is available in most countries,[1] employing undergraduate program catalogs enhances the adaptability of the TaP framework across various languages. For cases where an undergraduate program catalog is not available in a given language, we discuss alternative approaches for constructing the taxonomy in Appendix A.

Given the abundance of publicly available preference datasets in English and the relative scarcity of such datasets in other languages, we

[1] https://en.wikipedia.org/wiki/Undergraduate_education

apply the TaP framework to generate preference data in Chinese. Specifically, we adopt the `Undergraduate Program Catalog of Regular Higher Education Institutions of China`[2] as the guiding taxonomy for preference data generation. Manual inspection of the generated data reveals several desirable properties: (1) rich in knowledge, (2) high information density, and (3) broad coverage across domains and tasks. An illustrative example of a prompt and corresponding response is presented in Figure 4 of Appendix B to highlight these properties. We conduct both supervised fine-tuning and preference fine-tuning, including PPO and DPO, on LLMs using the TaP-generated data. Experimental results demonstrate that LLMs fine-tuned with TaP-generated data outperform those trained on existing open-source datasets. Our contributions can be summarized as follows:

- We propose the TaP framework, which enables automated, scalable preference data generation across various languages and offers fine-grained control over data composition.
- We apply TaP to generate high-quality preference data in Chinese, which can be used for both supervised and preference fine-tuning (e.g., PPO, DPO, GRPO). Additionally, the annotations produced by LLMs during the data generation process can be used to fine-tune open-source LLMs as judges (Wang et al., 2024b; Kim et al., 2024; Zhu et al., 2025), thereby reducing reliance on costly proprietary APIs and the risk of data leakage.
- We conduct extensive experiments using TaP-generated preference data to train five open-source LLMs ranging in size from 3B to 14B parameters, across three model families: LLaMA-3.1, Qwen2.5, and Gemma-2. The experimental results demonstrate that models trained on TaP-generated data outperform those trained on existing open-source datasets.

2 Related Work

Our research is closely related to studies on synthetic data generation using LLMs. Pioneering studies in this field have leveraged the strong in-context learning capabilities of LLMs by providing several demonstrations to guide the generation of synthetic samples (Honovich et al., 2023; Wang et al., 2023), typically requiring a small set of seed samples. Following this line of research, the follow up research usually necessitating a small number of seed samples to guide the LLMs in producing synthetic samples (Xu et al., 2024; Mukherjee et al., 2023; Li et al., 2024; Yue et al., 2024).

In contrast, TaP employs a taxonomy to guide LLMs in autonomously generating synthetic data, eliminating the need for seed samples authored by humans or LLMs. While TaP shares the goal of constructing preference datasets with UltraFeedback (Cui et al., 2024) and the HelpSteer series (Wang et al., 2024d,c), it differs in two key aspects: (1) TaP can produce prompts from scratch without relying on existing instruction datasets. This enables fine-grained control over dataset composition and enhances its applicability across various languages. Conversely, Cui et al. (2024) and Wang et al. (2024d,c) construct preference datasets by collecting prompts from existing instruction datasets. (2) TaP is applied to construct preference datasets in Chinese, addressing the scarcity of such data in that language. In comparison, Cui et al. (2024) and Wang et al. (2024d,c) primarily focus on constructing preference datasets in English.

3 Preference Dataset Curation

The TaP framework for automated and scalable preference data generation comprises four key steps: (1) **Taxonomy Specification**, wherein a structured taxonomy is constructed to comprehensively cover the common use cases of LLMs; (2) **Prompt Generation**, wherein LLMs are guided to produce prompts corresponding to the categories defined in the taxonomy; (3) **Response Generation and Selection**, wherein various LLMs generate responses to each prompt, and a subset is selected to balance annotation cost and response diversity; and (4) **Fine-grained Preference Annotation**, wherein LLMs evaluate and score the selected responses across multiple dimensions. Figure 1 illustrates the entire process of constructing the preference dataset using the TaP framework.

3.1 Taxonomy Specification

We adopt the `Undergraduate Program Catalog of Regular Higher Education Institutions of China` as the taxonomy to guide LLMs in generating preference data in Chinese. This catalog comprehensively lists all undergraduate sub-

[2] `http://www.moe.gov.cn/srcsite/A08/moe_1034/s4930/202403/W020240319305498791768.pdf`

jects offered by Chinese universities, encompassing a diverse range of academic disciplines, spanning humanities, social sciences, and STEM fields. It comprises 816 distinct subjects, systematically grouped into 93 discipline categories. These categories are further organized into 12 primary degree-conferring disciplines, ensuring a structured organization of undergraduate knowledge.

To construct preference datasets that cover diverse domains and tasks, we have incorporated all subjects specified in the catalog into our taxonomy, with the exception of those in the Foreign Languages and Literatures discipline, which includes numerous low-resource languages. Instead, we have selectively included 12 subjects: the top nine languages, along with Linguistics, Translation, and Business English. Consequently, our taxonomy encompasses a total of 724 subjects, consisting of 12 from the Foreign Languages and Literatures discipline and 712 from other disciplines.

3.2 Prompt Generation

We adopt a multi-stage prompting strategy, rather than instructing LLMs to generate prompts directly, to enhance both the diversity and coverage of the resulting outputs. **In the first stage**, LLMs are prompted to generate diverse question types and brief descriptions for each subject within a predefined taxonomy through a three-round interaction. Each round produces a distinct set of question types and descriptions, thereby increasing both the quantity and diversity of the generated outputs. The prompt used in this stage is provided in Figure 5 of Appendix C.1. **In the second stage**, LLMs refine the question type descriptions from the first stage to improve their clarity and real-world relevance. This refinement aims to enhance the overall quality of the final prompts. The prompt used in this stage is shown in Figure 6 of Appendix C.1. **In the third stage**, we employ a two-round interaction to address the issue of LLMs occasionally producing incomplete prompts lacking essential contextual information.[3] In the first round, LLMs generate a candidate prompt. In the second round, they assess whether the prompt contains all necessary information. If a prompt is identified as incomplete, the LLMs regenerate it to ensure completeness. The prompt used for this interaction is illustrated in Figure 7 of Appendix C.1. Additionally, we find that LLMs sometimes generate prompts beyond their capabilities, such as those requiring real-world actions. To mitigate this issue, we employ the LLM as a checker to evaluate whether a given prompt is feasible within its operational constraints. If a prompt is deemed infeasible, it is revised, with up to three regeneration attempts. The evaluation prompt for this step is presented in Figure 8 of Appendix C.1. To verify the reliability of LLMs in identifying such infeasible prompts, we manually reviewed a subset of evaluations generated by GPT-4. The results demonstrate that GPT-4 reliably identifies prompts that exceed its capabilities. Further details are provided in Appendix C.2.

[3]For instance, LLMs may generate prompts that request a summary of an article without providing the article itself.

3.3 Response Generation and Selection

We select 65 widely used LLMs, ranging in size from 0.5B to 236B parameters, to generate responses for each prompt, thereby enhancing the diversity and coverage of LLM-generated responses. The models span both dense and MoE architectures and include both proprietary and open-source models. The rationale for model selection, along with detailed specifications of the chosen LLMs, is provided in Appendix C.3. Each model is prompted using the system prompt shown in Figure 9 of Appendix C.1, resulting in 65 responses per prompt.

Given that each prompt produces 65 responses, leading to 2,080 pairwise preference pairs to be annotated per prompt. To manage dataset size and reduce annotation costs while preserving response diversity and coverage, we apply K-means clustering to group the responses for each prompt into five clusters. From each cluster, we select a representative response. Specifically, we choose the response closest to the cluster centroid, except in the case of the cluster containing the response generated by GPT-4. In this case, we instead select the GPT-4 response to ensure the inclusion of at least one high-quality response per prompt.

3.4 Fine-grained Preference Annotation

We employ multiple LLMs to conduct preference annotation for each pairwise combination of selected responses to a given prompt. Specifically, we utilize GPT-4 alongside five high-performing open-source LLMs: Qwen2-72B-Instruct, Command R+, Mistral-Large-Instruct-2407, Llama-3.1-70B-Instruct, and Deepseek V2.5.[4]

[4]Due to the high cost of using GPT-4 for preference annotation, it is used only to annotate pairwise responses to prompts it generates. In contrast, annotations for responses to DeepSeek-V2 prompts are handled by open-source LLMs.

Dataset	#Samples	#Prompts	Prompt Length	Response Length	Fine-grained Annotation?	Annotation Type	Annotator	Languages
Huozi-RLHF (Huozi-Team, 2024)	16,918	3,725	69.19 ± 107.38	216.02 ± 165.44	✗	Ranking	Human	Chinese
Chinese-DPO-Pairs	10,735	10,717	220.23 ± 318.37	401.90 ± 347.63	✗	Ranking	AI	Chinese
TaP (GPT-4)	261,874	27,715	379.29 ± 159.11	982.89 ± 919.97	✓	Scalar	AI	Chinese

Table 1: Comparison of open-source preference datasets in Chinese with the dataset constructed using TaP. "TaP (GPT-4)" indicates that the prompts in the dataset were generated by GPT-4. The column labeled "#Prompts" represents the number of unique prompts in each dataset. "Prompt Length" and "Response Length" are expressed in the format $XX \pm YY$, where XX denotes the mean number of tokens and YY represents the standard deviation. Tokenization was performed using the `cl100k_base` encoding provided by the tiktoken library.

For preference annotation, we adopt a pairwise comparison approach, wherein LLMs are prompted to choose the preferred response between two alternatives, rather than assigning independent scores to each response as in pointwise grading. Our preliminary experiments indicate that pointwise grading often fails to distinguish subtle differences between similar responses, frequently assigning them identical scores. Despite its higher annotation cost, pairwise comparison is employed to obtain more precise preference annotations. In this approach, we present two responses to LLMs and ask them to score each one across four dimensions: relevance, correctness, clarity, and completeness. These dimensions facilitate a comprehensive assessment of response quality. The resulting scores can be used to train fine-grained reward models, which in turn enhance control over LLM-generated responses by assigning weights to rewards from different fine-grained reward models during RLHF (Wu et al., 2023). The prompt used for pairwise comparison is provided in Figure 10 of Appendix C.1. Additionally, to mitigate position bias during pairwise comparison (Wang et al., 2024a), we adopt the Balanced Position Calibration strategy, which involves conducting an additional comparison with the positions of the two responses swapped.

To derive preference labels from the pairwise comparison scores, we compute an overall score for each response in a pair. The response with the higher overall score is labeled as preferred. To compute each overall score, we first average the scores for each evaluation dimension across both positions to obtain calibrated scores. These calibrated scores are then equally weighted and averaged to produce a final score for each response. Although domain-specific applications might prioritize certain dimensions, for example, in medical applications, correctness may be prioritized, we assign equal weight to all four dimensions to support general-purpose LLM training.

4 Dataset Statistics

We employ GPT-4 and DeepSeek-V2 to generate prompts corresponding to the 724 subjects defined in our taxonomy, which is derived from the `Undergraduate Program Catalog of Regular Higher Education Institutions of China`. GPT-4 generates a total of 27,715 prompts, while DeepSeek-V2 produces 50,929 prompts. The distributions of these prompts across the 12 primary disciplines in the taxonomy are presented in Figure 2 and Figure 3 in Appendix D, respectively. Since some prompts need to be regenerated after failing the feasibility check, a single question type may correspond to multiple prompts. As a result, GPT-4 yields 19,840 distinct question types, and DeepSeek-V2 yields 24,839 distinct question types.

For constructing the preference dataset, we rely solely on preference annotations generated by GPT-4, given its pioneering performance.[5] Table 1 presents a comparative summary of the preference dataset constructed by TaP and existing open-source preference datasets. Compared to these existing datasets, the TaP-constructed dataset encompasses a larger number of samples and prompts, features longer prompts and responses, and provides fine-grained annotations.

5 Experiments

To assess the effectiveness of TaP and to evaluate the quality of the preference datasets it constructs, we conducted both supervised fine-tuning and preference fine-tuning, including PPO and DPO, on LLMs using the datasets generated by TaP.

5.1 Supervised Fine-tuning

To conduct supervised fine-tuning using the preference data generated by TaP, we utilized the prompts

[5]While incorporating annotations from multiple LLMs might improve dataset quality, we leave the integration of annotations from other open-source LLMs for future work.

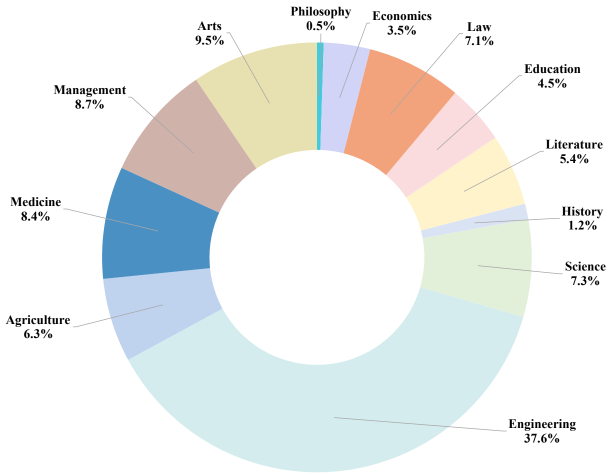

Figure 2: Distribution of prompts generated by GPT-4 across the 12 primary disciplines in the taxonomy.

Dataset	#Samples	Languages
Firefly	1,649,399	Chinese
Alpaca-GPT-4-ZH (Peng et al., 2023)	48,818	Chinese
COIG (Zhang et al., 2023a)	178,246	Chinese
MOSS-SFT (Sun et al., 2024)	1,074,551	English, Chinese
COIG-CQIA (Bai et al., 2024)	44,694	Chinese
Infinity-Instruct	757,938	Chinese
BELLE-SFT (BELLEGroup, 2023)	3,606,402	Chinese
TaP-SFT (GPT-4)	19,840	Chinese
TaP-SFT (DeepSeek-V2)	24,839	Chinese

Table 2: Comparison of open-source datasets containing a substantial number of samples in Chinese for supervised fine-tuning with datasets derived from preference datasets constructed by TaP. "TaP-SFT (GPT-4)" and "TaP-SFT (DeepSeek-V2)" indicate that the prompts were generated by GPT-4 and DeepSeek-V2. For the Infinity-Instruct dataset, which includes samples in multiple languages, Chinese samples were selected based on the language tags provided within the data.

from the dataset along with GPT-4-generated responses. To ensure a balanced distribution of question types, we randomly selected one prompt and its corresponding response for each question types with multiple prompts. The resulting dataset comprises 19,840 samples generated by GPT-4 and 24,839 samples produced by DeepSeek-V2. A comparative summary of this supervised fine-tuning dataset, alongside other open-source Chinese datasets, is provided in Table 2.

We conducted supervised fine-tuning on five open-source LLMs: (1) Qwen2.5-3B, (2) Qwen2.5-7B, (3) Llama-3.1-8B, (4) Gemma-2-9B, and (5) Qwen2.5-14B. These LLMs span three model families, with parameter sizes ranging from 3B to 14B.

5.2 Preference Fine-tuning

We adopt two dominant approaches for preference fine-tuning of LLMs using our constructed preference dataset: PPO and DPO. To facilitate comparison with other open-source preference datasets, we utilize Huozi-RLHF (Huozi-Team, 2024) and Chinese-DPO-Pairs[6] for preference fine-tuning. Further details regarding the experimental setup for PPO and DPO can be found in Appendix E.1.

5.3 Evaluation

To evaluate the performance of trained LLMs, we employ MT-Bench-zh[7] as the validation set, a Chinese translation of MT-Bench (Zheng et al., 2023). For testing, we adopt AlignBench (Liu et al., 2024). Both benchmarks rely on high-performing LLMs, such as GPT-4o, to evaluate model responses. Accordingly, we report scores generated by both GPT-4o and DeepSeek-V3 to provide a comprehensive evaluation. Further details regarding the evaluation setup are provided in Appendix E.2.

[6] `https://huggingface.co/datasets/wenbopan/Chinese-dpo-pairs`

[7] `https://github.com/HIT-SCIR/huozi/tree/main/data/mt-bench-zh`

6 Experimental Results

We present the aggregated experimental results for supervised fine-tuning, DPO, and PPO in Table 3, 4, and 6 of Appendix F, respectively. The corresponding detailed experimental results are provided in Tables 7 through 15 in Appendix F. When trained on the same dataset, we observe that larger LLMs consistently outperform their smaller counterparts within the Qwen2.5 series. Moreover, LLMs trained on data constructed using TaP consistently demonstrate strong performance across supervised fine-tuning, DPO, and PPO. This finding underscores the effectiveness of TaP in generating high-quality training data for both supervised and preference fine-tuning of LLMs. Furthermore, the experimental results yield additional findings:

To ensure optimal model performance when scaling up data for supervised fine-tuning, it is crucial to consider factors such as data quality, diversity, and coverage. The experimental results presented in Table 3 suggest that despite the relatively small volume of `TaP-SFT (GPT-4)`, LLMs trained on this dataset outperform those trained on other open-source datasets. **Notably, LLMs trained on `TaP-SFT (GPT-4)` consistently surpass those trained on `BELLE-SFT`, even**

Model	Dataset	AlignBench						MT-Bench-zh	
		GPT-4o			DeepSeek-V3				
		Reasoning	Language	Overall	Reasoning	Language	Overall	GPT-4o	DeepSeek-V3
Qwen2.5-3B	Firefly	3.22	4.34	3.78	3.38	4.37	3.87	4.41	4.45
	Alpaca-GPT-4-ZH (Peng et al., 2023)	4.16	4.87	4.51	3.99	4.75	4.37	4.83	4.68
	COIG (Zhang et al., 2023a)	2.73	4.12	3.42	2.84	4.03	3.43	3.35	3.44
	MOSS-SFT (Sun et al., 2024)	3.82	4.96	4.39	3.73	4.82	4.27	4.66	4.38
	COIG-CQIA (Bai et al., 2024)	3.61	4.38	3.99	3.65	4.31	3.98	4.47	4.35
	Infinity-Instruct	4.02	5.11	4.57	3.91	4.92	4.41	4.89	4.68
	BELLE-SFT (BELLEGroup, 2023)	3.88	5.11	4.49	3.82	4.99	4.41	4.65	4.49
	TaP-SFT (GPT-4)	4.99	**5.85**	5.42	4.57	**5.46**	5.02	**5.79**	5.53
	TaP-SFT (DeepSeek-V2)	**5.25**	5.65	**5.45**	**4.89**	5.39	**5.14**	5.67	**5.60**
Qwen2.5-7B	Firefly	3.45	4.61	4.03	3.62	4.64	4.13	4.43	4.47
	Alpaca-GPT-4-ZH (Peng et al., 2023)	4.90	5.31	5.10	4.53	5.14	4.83	5.02	4.89
	COIG (Zhang et al., 2023a)	2.82	4.41	3.62	3.11	4.38	3.75	3.48	3.66
	MOSS-SFT (Sun et al., 2024)	4.71	5.26	4.98	4.55	5.17	4.86	5.06	4.90
	COIG-CQIA (Bai et al., 2024)	4.72	5.13	4.92	4.51	5.02	4.76	4.73	4.51
	Infinity-Instruct	4.92	5.63	5.28	4.70	5.54	5.12	4.96	4.95
	BELLE-SFT (BELLEGroup, 2023)	4.32	5.44	4.88	4.16	5.35	4.76	4.93	4.83
	TaP-SFT (GPT-4)	**6.62**	**6.27**	**6.45**	**6.10**	5.83	**5.96**	**6.16**	**5.99**
	TaP-SFT (DeepSeek-V2)	6.25	6.07	6.16	5.77	**5.93**	5.85	6.10	5.97
Llama-3.1-8B	Firefly	2.40	3.97	3.19	2.40	4.07	3.24	3.72	3.69
	Alpaca-GPT-4-ZH (Peng et al., 2023)	2.77	4.28	3.53	2.64	4.12	3.38	4.21	4.28
	COIG (Zhang et al., 2023a)	1.79	2.98	2.38	1.88	3.04	2.46	2.68	2.74
	MOSS-SFT (Sun et al., 2024)	2.84	4.41	3.63	2.69	4.32	3.50	4.21	4.06
	COIG-CQIA (Bai et al., 2024)	2.54	3.57	3.06	2.39	3.56	2.97	3.32	3.34
	Infinity-Instruct	3.15	5.07	4.11	3.01	**4.95**	3.98	4.71	4.60
	BELLE-SFT (BELLEGroup, 2023)	3.47	5.05	4.26	3.24	4.88	4.06	4.63	4.74
	TaP-SFT (GPT-4)	**3.87**	**5.14**	**4.50**	3.40	4.73	**4.06**	**5.24**	**4.95**
	TaP-SFT (DeepSeek-V2)	3.61	4.75	4.18	**3.47**	4.61	4.04	5.11	4.89
Gemma-2-9B	Firefly	2.36	3.95	3.15	2.44	4.06	3.25	3.92	3.77
	Alpaca-GPT-4-ZH (Peng et al., 2023)	3.17	4.22	3.70	3.19	4.17	3.68	4.39	4.09
	COIG (Zhang et al., 2023a)	1.70	2.81	2.25	1.95	2.80	2.38	2.69	2.65
	MOSS-SFT (Sun et al., 2024)	2.84	4.21	3.53	2.73	4.19	3.46	4.14	4.23
	COIG-CQIA (Bai et al., 2024)	2.69	3.58	3.14	2.64	3.56	3.10	3.36	3.13
	Infinity-Instruct	3.72	**5.21**	4.47	3.57	4.99	4.28	4.78	4.45
	BELLE-SFT (BELLEGroup, 2023)	3.72	5.16	4.44	**3.68**	**5.07**	**4.37**	4.69	4.87
	TaP-SFT (GPT-4)	**4.10**	5.17	**4.64**	3.60	4.88	4.24	**5.37**	**5.03**
	TaP-SFT (DeepSeek-V2)	3.96	4.90	4.43	3.61	4.58	4.10	4.90	4.69
Qwen2.5-14B	Firefly	3.94	4.94	4.44	4.10	4.89	4.49	4.77	4.79
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.64	5.73	5.68	5.34	5.73	5.54	5.44	5.36
	COIG (Zhang et al., 2023a)	3.30	4.84	4.07	3.38	4.86	4.12	3.63	3.56
	MOSS-SFT (Sun et al., 2024)	5.25	5.59	5.42	4.96	5.48	5.22	5.26	5.04
	COIG-CQIA (Bai et al., 2024)	5.14	5.27	5.20	4.94	5.26	5.10	4.92	4.78
	Infinity-Instruct	5.26	5.94	5.60	5.16	5.81	5.48	5.33	5.06
	BELLE-SFT (BELLEGroup, 2023)	4.70	5.70	5.20	4.55	5.61	5.08	4.99	4.83
	TaP-SFT (GPT-4)	**6.76**	**6.51**	**6.64**	6.27	**6.27**	6.27	**6.34**	6.18
	TaP-SFT (DeepSeek-V2)	6.34	5.92	6.13	**6.46**	6.24	**6.35**	6.12	**6.19**

Table 3: Performance comparison of five LLMs on AlignBench and MT-Bench-zh after supervised fine-tuning with various datasets. "GPT-4o" and "DeepSeek-V3" indicate that the responses for the respective benchmarks were evaluated using GPT-4o and DeepSeek-V3, respectively.

though `BELLE-SFT` contains over 180 times the number of samples as `TaP-SFT (GPT-4)`. This finding indicates that, beyond dataset size, factors such as data quality, diversity, and coverage play a critical role in model performance. These results align with the findings of LIMA (Zhou et al., 2023), which demonstrate that LLMs trained on just 1,000 carefully curated samples can achieve remarkably strong performance. However, LIMA's heavy reliance on human curation poses challenges for scalability. In contrast, TaP necessitates human involvement only in designing the taxonomy,[8] making it significantly more scalable.

The capabilities of pre-trained LLMs serve as a critical foundation for the performance of models subsequently trained through supervised fine-tuning or preference fine-tuning. The experimental results presented in Tables 3, 4, and 6 of Appendix F indicate that LLMs initialized from Qwen2.5-3B usually outperform those initialized from Llama-3.1-8B and Gemma-2-9B, despite the latter two LLMs having nearly three times the number of parameters as Qwen2.5-3B, even when trained on the same dataset. We hypothesize that this performance gap arises from the limited volume of Chinese corpora used in the pre-training of Llama-3.1-8B and Gemma-2-9B, which constrains their ability to generate high-quality responses in Chinese after undergoing supervised fine-tuning or preference fine-tuning. In contrast, the Qwen2.5 series of LLMs have been extensively trained on Chinese corpora, thereby enhancing their capacity for instruction following and alignment with human preferences and values after fine-tuning.

[8] Instead of constructing the taxonomy manually, it can alternatively be constructed either by LLMs alone or through collaboration between humans and LLMs.

Model	Dataset	AlignBench						MT-Bench-zh	
		GPT-4o			DeepSeek-V3				
		Reasoning	Language	Overall	Reasoning	Language	Overall	GPT-4o	DeepSeek-V3
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.02	5.23	4.63	3.88	5.04	4.46	5.03	4.94
	Chinese-DPO-Pairs	4.46	5.52	4.99	4.20	5.24	4.72	5.04	4.95
	TaP (GPT-4)	**5.19**	**5.90**	**5.55**	**4.79**	**5.62**	**5.21**	**5.88**	**5.63**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	5.15	5.82	5.49	4.65	5.50	5.08	5.72	5.24
	Chinese-DPO-Pairs	5.25	6.01	5.63	**4.90**	5.51	5.20	5.91	5.68
	TaP (GPT-4)	**5.57**	**6.06**	**5.81**	**4.90**	**5.65**	**5.28**	**6.03**	**5.79**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.53	5.39	4.96	4.37	5.34	4.85	5.25	4.98
	Chinese-DPO-Pairs	4.88	5.53	5.21	4.64	5.40	5.02	5.23	5.23
	TaP (GPT-4)	**5.11**	**5.74**	**5.43**	**4.87**	**5.58**	**5.22**	**5.54**	**5.38**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.24	6.34	6.29	5.83	6.00	5.92	6.10	5.85
	Chinese-DPO-Pairs	6.49	6.42	6.46	5.94	6.09	6.01	6.16	5.98
	TaP (GPT-4)	**6.59**	**6.66**	**6.62**	**6.03**	**6.18**	**6.11**	**6.36**	**6.11**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.52	5.19	4.36	3.29	5.06	4.17	4.85	4.58
	Chinese-DPO-Pairs	3.53	5.57	4.55	3.29	5.29	4.29	4.94	4.88
	TaP (GPT-4)	**3.96**	**5.62**	**4.79**	**3.50**	**5.37**	**4.43**	**5.41**	**5.14**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.70	5.22	4.46	3.33	4.86	4.09	5.19	4.84
	Chinese-DPO-Pairs	4.07	5.36	4.71	**3.59**	**5.07**	**4.33**	5.30	4.84
	TaP (GPT-4)	**4.10**	**5.50**	**4.80**	3.52	4.99	4.25	**5.59**	**5.16**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.93	5.22	4.58	3.54	5.05	4.29	4.77	4.62
	Chinese-DPO-Pairs	3.86	5.32	4.59	3.61	5.21	4.41	4.91	4.84
	TaP (GPT-4)	**4.55**	**5.90**	**5.22**	**4.09**	**5.44**	**4.76**	**5.54**	**5.23**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	4.18	5.23	4.70	3.77	4.93	4.35	5.39	5.01
	Chinese-DPO-Pairs	4.67	5.52	5.09	**4.09**	**5.21**	**4.65**	5.54	5.20
	TaP (GPT-4)	**4.72**	**5.61**	**5.16**	4.05	5.14	4.60	**5.78**	**5.33**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	5.75	5.90	5.82	5.64	5.81	5.73	5.41	5.48
	Chinese-DPO-Pairs	5.85	5.96	5.90	5.47	5.79	5.63	5.54	5.58
	TaP (GPT-4)	**6.44**	**6.39**	**6.42**	**6.05**	**6.17**	**6.11**	**5.84**	**5.89**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.94	6.64	6.79	6.41	6.28	6.35	6.25	6.11
	Chinese-DPO-Pairs	6.99	6.85	6.92	6.41	6.43	6.42	6.34	6.14
	TaP (GPT-4)	**7.27**	**7.08**	**7.17**	**6.52**	**6.52**	**6.52**	**6.66**	**6.35**

Table 4: Performance comparison of LLMs trained with DPO using different datasets on AlignBench and MT-Bench-zh. The model names include two possible suffixes: "Open" and "TaP." The "Open" suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas "TaP" denotes initialization from models trained on a dataset constructed by TaP. Additionally, "GPT-4o" and "DeepSeek-V3" specify that responses were evaluated using GPT-4o and DeepSeek-V3, respectively.

The effectiveness of applying supervised fine-tuning and preference fine-tuning using the same training dataset varies across different LLMs. The experimental results presented in Tables 3, 4, and 6 of Appendix F indicate that the optimal open-source datasets for fine-tuning vary across LLMs. For instance, as shown in Table 3, the open-source fine-tuning dataset that yields the best performance for Qwen2.5-7B on MT-Bench-zh, when GPT-4o serves as the judge, is MOSS-SFT. However, for Llama-3.1-8B, the open-source dataset that results in the best performance under the same evaluation conditions is Infinity-Instruct. These findings align with the hypothesis proposed by Zhang et al. (2025), which posits that supervised fine-tuning datasets that are more closely aligned with the distribution of pre-trained LLMs tend to be more effective. Nevertheless, LLMs trained on data constructed by TaP consistently achieve strong performance and generally outperform those trained on other open-source datasets. This demonstrates that data constructed by TaP can be effectively employed for both supervised fine-tuning and preference fine-tuning across various LLMs.

7 Conclusion

In this paper, we propose the TaP framework to facilitate automated and scalable construction of preference datasets across various languages. Grounded in a structured taxonomy, TaP offers fine-grained control over dataset composition, ensuring both diversity and comprehensive coverage. To evaluate the effectiveness of TaP and address the scarcity of preference datasets in Chinese, we apply TaP to construct preference datasets in Chinese, using a taxonomy derived from the `Undergraduate Program Catalog of Regular Higher Education Institutions of China`. We conduct supervised fine-tuning and preference fine-tuning on LLMs ranging from 3B to 14B parameters across three model families, utilizing the TaP-generated datasets. Experimental results show that LLMs trained on TaP-generated datasets outperform those trained on open-source datasets.

Limitations

A key limitation of our study is the absence of extensive human evaluation to assess the quality of the data generated by TaP. While we acknowledge that incorporating human evaluation could further validate the effectiveness of the TaP framework and enhance the quality of the synthetic data, our efforts were limited to a small-scale human evaluation focused on the reliability of LLMs in identifying instructions beyond their capabilities (see Appendix C.2 for details). However, the volume of data produced by TaP makes comprehensive human evaluation prohibitively resource-intensive. Moreover, human evaluation is time-consuming, labor-intensive, and susceptible to subjective biases and inconsistencies (Hosking et al., 2024). As a result, we did not conduct extensive human evaluation of the TaP-generated data. Nevertheless, the strong performance of LLMs trained on TaP-generated data can serve as an indirect proxy of data quality (Wang et al., 2023; Xu et al., 2024; Ding et al., 2023). This performance is therefore interpreted as indirect evidence supporting the acceptable quality of the synthetic data produced by TaP.

Another limitation of our work is that previous research has highlighted the risk of model collapse when training LLMs on synthetic data (Gerstgrasser et al., 2024; Kazdan et al., 2024; Shumailov et al., 2024). Accordingly, LLMs trained on datasets constructed using TaP may also be susceptible to similar risks.

References

Yuelin Bai, Xinrun Du, Yiming Liang, Yonggang Jin, Ziqiang Liu, Junting Zhou, Tianyu Zheng, Xincheng Zhang, Nuo Ma, Zekun Wang, Ruibin Yuan, Haihong Wu, Hongquan Lin, Wenhao Huang, Jiajun Zhang, Wenhu Chen, Chenghua Lin, Jie Fu, Min Yang, Shiwen Ni, and Ge Zhang. 2024. COIG-CQIA: quality is all you need for chinese instruction fine-tuning. *CoRR*, abs/2403.18058.

BELLEGroup. 2023. Belle: Be everyone's large language model engine. https://github.com/LianjiaTech/BELLE.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, 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, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, 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, 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, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, 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, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *CoRR*, abs/2404.01413.

Sian Gooding and Hassan Mansoor. 2023. The impact of preference agreement in reinforcement learning from human feedback: A case study in summarization. *CoRR*, abs/2311.04919.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *CoRR*, abs/2306.11644.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14,* *2023*, pages 14409–14428. Association for Computational Linguistics.

Tom Hosking, Phil Blunsom, and Max Bartolo. 2024. Human feedback is not gold standard. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Huozi-Team. 2024. Huozi: Leveraging large language models for enhanced open-domain chatting. `https://github.com/HIT-SCIR/huozi`.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hanna Hajishirzi. 2024. Unpacking DPO and PPO: disentangling best practices for learning from preference feedback. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Joshua Kazdan, Rylan Schaeffer, Apratim Dey, Matthias Gerstgrasser, Rafael Rafailov, David L. Donoho, and Sanmi Koyejo. 2024. Collapse or thrive? perils and promises of synthetic data in a self-generating world. *CoRR*, abs/2410.16713.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 4334–4353. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of RLHF on LLM generalisation and diversity. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need II: phi-1.5 technical report. *CoRR*, abs/2309.05463.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. Alignbench: Benchmarking chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11621–11640. Association for Computational Linguistics.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *CoRR*, abs/2306.02707.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7737–7746. Association for Computational Linguistics.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross J. Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nat.*, 631(8022):755–759.

Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, Lingling Wu, Zhangyue Yin, Xuanjing Huang, Yu-Gang Jiang, and Xipeng Qiu. 2024. Moss: An open conversational large language model. *Machine Intelligence Research*, 21(5):888–905.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9440–9450. Association for Computational Linguistics.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024b. Pandalm: An automatic evaluation benchmark for LLM instruction tuning optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024c. Helpsteer2: Open-source dataset for training top-performing reward models. *CoRR*, abs/2406.08673.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2024d. Helpsteer: Multi-attribute helpfulness dataset for steerlm. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3371–3384. Association for Computational Linguistics.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Xiang Yue, Tianyu Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Dylan Zhang, Qirun Dai, and Hao Peng. 2025. The best instruction-tuning data are those that fit. *Preprint*, arXiv:2502.04194.

Ge Zhang, Yemin Shi, Ruibo Liu, Ruibin Yuan, Yizhi Li, Siwei Dong, Yu Shu, Zhaoqun Li, Zekun Wang, Chenghua Lin, Wenhao Huang, and Jie Fu. 2023a. Chinese open instruction generalist: A preliminary release. *CoRR*, abs/2304.07987.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023b. Instruction tuning for large language models: A survey. *CoRR*, abs/2308.10792.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. Lmsys-chat-1m: A large-scale real-world LLM conversation dataset.

In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. Judgelm: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

A Alternative Approaches for Constructing the Taxonomy

In cases where the undergraduate program catalog is unavailable, the undergraduate program catalogs from high-resource languages can serve as representative taxonomies to guide the construction of analogous taxonomies for other languages. Furthermore, we emphasize that using the undergraduate program catalog as the taxonomy represents a specific and illustrative case of applying the TaP framework for synthetic data generation. The taxonomy employed by the TaP framework is not limited to the undergraduate program catalog. An alternative approach to constructing the taxonomy involves analyzing and categorizing the potential use cases of LLMs, a method that can be extended to other languages.

While comprehensively enumerating and categorizing all possible use cases of LLMs is inherently challenging, a practical strategy is to collect user–model interactions and analyze these conversations to systematically identify and categorize use cases (Zhao et al., 2024; Zheng et al., 2024). For low-resource languages with limited resources, a feasible approach would be to initially deploy a LLM trained on all available data for these languages, potentially supplemented with data translated from high-resource languages. Following deployment, user–model conversations can be continuously collected and analyzed to construct the taxonomy grounded in real-world use cases. Additionally, both the taxonomy and the LLMs themselves can be incrementally updated as sufficient new user-model conversations become available, thereby improving model performance over time.

B An Illustrative Example

Figure 4 presents an example prompt and its corresponding response from the preference dataset constructed using TaP. Both the prompt and the response were generated by GPT-4.

C Dataset Construction

C.1 Prompts

The prompts utilized in the data construction process are depicted in Figures 5 through 10.

C.2 Human Evaluation on Feasibility Check

To evaluate the reliability of LLMs in identifying instructions that exceed their capabilities, we randomly sampled 100 instructions along with the corresponding assessments generated by GPT-4 indicating whether each instruction was beyond the model's capabilities. Sampling was performed using NumPy with a fixed random seed of 42 to ensure reproducibility.

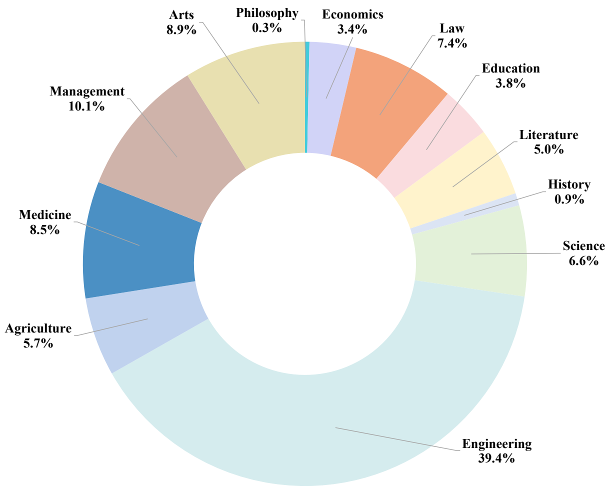

Figure 3: Distribution of prompts generated by DeepSeek-V2 across the 12 primary disciplines in the taxonomy.

According to GPT-4's assessments, 71 of the 100 sampled instructions were classified as within the model's capabilities, while the remaining 29 were identified as exceeding them. We then manually verified the correctness of these assessments. Notably, we found that 99 out of the 100 assessments were correct. This result demonstrates GPT-4's high reliability in recognizing its own limitations when explicitly prompted with instructions that incorporate descriptions of its key limitations.

For illustrative purposes, Figure 11 presents a representative example from the sampled set, comprising both the instruction and GPT-4's corresponding assessment. In this case, GPT-4 correctly identified the instruction as exceeding the capabilities of LLMs.

C.3 LLMs for Response Generation

To enhance the diversity and coverage of responses generated by the selected models, we aimed to include a broad spectrum of LLM families and architectures. Given the substantial computational and time costs associated with inference across such a large number of models, inference efficiency was prioritized in selecting open-source models. Specifically, aside from proprietary models such as GPT-4, inference efficiency served as the primary criterion for open-source model selection.

To facilitate efficient and scalable inference, we utilized the vLLM library,[9] which is known for its high throughput and memory efficiency in LLM inference. While vLLM supports a wide range of models, its compatibility is primarily limited to widely adopted models. Consequently, our selection was constrained to models supported by version 0.6.1.post2 of vLLM, which was the version used to generate responses for all prompts.

From the set of models compatible with this version of vLLM, we selected those that met at least one of the following criteria: (1) The model demonstrated exceptional performance across various benchmarks and attained significant popularity in academic and industrial communities (e.g., models from the LLaMA, Qwen, and Gemma families); (2) The model was trained on extensive Chinese corpora, thereby ensuring the generation of high-quality synthetic Chinese data.

Based on these criteria, we selected 63 open-source LLMs. For the remaining two models, GPT-4 (proprietary) and DeepSeek-V2 (unsupported by the vLLM version we used), we employed their official APIs to generate responses. Table 5 presents a detailed summary of the architectures and parameter sizes of all LLMs used for response generation.

D Dataset Statistics

Figure 3 illustrates the distribution of prompts generated by DeepSeek-V2 across the 12 primary disciplines defined in the taxonomy.

E Experiments

E.1 Preference Fine-tuning

For all preference datasets used in our experiments, we randomly split each dataset into two equal subsets, using one for training reward models and the other for PPO and DPO training.

For PPO and DPO training, we initialize the policies using checkpoints from supervised fine-tuning of five open-source pre-trained LLMs. Specifically, we select the best-performing checkpoint trained on two distinct types of training data for each open-source pre-trained LLM, based on their performance on the validation dataset: (1) the highest-performing checkpoint among LLMs trained on seven open-source supervised fine-tuning datasets and (2) the highest-performing checkpoint obtained from an LLM trained on supervised fine-tuning datasets derived from our constructed preference dataset. Consequently, we conduct PPO and DPO training on a total of twelve checkpoints.

For PPO training, we first train the reward model to score the responses generated by the policy. Following common practice, we initialize the reward models from pre-trained LLMs (Gooding and Mansoor, 2023; Cui et al., 2024; Kirk et al., 2024). Specifically, the reward models are initialized from the same pre-trained LLMs used to initialize the LLMs for supervised fine-tuning, with the exception that the reward model for the policy initialized from the checkpoints derived from supervised fine-tuning of Qwen2.5-3B is initialized from Qwen2.5-14B. Additionally, we constrain the rewards produced by the reward models to a predefined range $[-R, R]$ using an auxiliary loss function to stabilize PPO training, where R is a positive constant. Given a prompt x, a preferred response y_w and less preferred response y_l, the reward model is parameterized by ϕ, with $\mathcal{D}$ denoting the preference dataset used for training. The auxiliary loss function is formulated as follows:

$$
\begin{aligned}
\mathcal{L}_{\text{aux}}(\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[& \max(0, r_\phi(x, y_w) - R)^2 \\
& + \max(0, -r_\phi(x, y_w) - R)^2 \\
& + \max(0, r_\phi(x, y_l) - R)^2 \\
& + \max(0, -r_\phi(x, y_l) - R)^2 \Big]
\end{aligned} \tag{1}
$$

The auxiliary loss function is incorporated into the overall loss function for training the reward models using a hyperparameter λ that adjusts its weight. The resulting loss function for training reward models is defined as follows:

$$
\begin{aligned}
& \mathcal{L}_{\text{RM}}(\phi) = \\
& -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right) + \lambda\mathcal{L}_{\text{aux}}(\phi)\Big]
\end{aligned} \tag{2}
$$

The hyperparameter λ is set to 0.05 when training reward models with our constructed preference dataset. However, due to the relatively smaller size of open-source preference datasets, we reduce λ to 0.01 when using them for training. Additionally, the hyperparameter R is fixed at 5, and all reward models are trained for one epoch, which we found sufficient for convergence.

For PPO training, we train the policy for two epochs when using our constructed dataset and

[9] https://github.com/vllm-project/vllm

for three epochs when using open-source datasets, given their relatively smaller size. We adjust the learning rate according to the `cosine` schedule with a warm-up phase of 50 steps, using a peak learning rate of 1e-6 for policy models and 5e-6 for critic models. Furthermore, we vary the KL reward coefficient within $\{0.01, 0.05\}$ based on the KL divergence between the policy and reference policy during PPO training.

For DPO training, we follow a similar strategy to PPO. Specifically, we train the policy for two epochs when using our constructed dataset and for three epochs when using open-source datasets. The learning rate follows the `cosine` schedule with a warm-up phase of 50 step and a peak learning rate of 2e-7. Following Ivison et al. (2024), we set β to 0.01 across all experiments.

E.2 Evaluation

For LLMs trained through supervised fine-tuning and DPO, we exclusively use scores from GPT-4o to select the best-performing checkpoint. Conversely, for models trained via supervised learning on `TaP-SFT (DeepSeek-V2)`, we rely on DeepSeek-V3 scores due to the significant API costs associated with GPT-4o. Similarly, for models trained using PPO, we also utilize DeepSeek-V3 scores for checkpoint selection.

F Experimental Results

The aggregated experimental results for PPO are presented in Table 6, while detailed results for supervised fine-tuning, PPO, and DPO are provided in Tables 7 through 15.

An Illustrative Example

Prompt

设计并编写一个程序以模拟智能海洋装备中的自动路径规划系统。该系统需要根据给定的海洋环境信息，为无人潜水器提供从起点到终点的最优路径。海洋环境信息包括但不限于海域的大 小、存在的障碍物（如珊瑚礁、深海沟），以及起点和终点的坐标。你需要采用适当的搜索算法或路径规划算法（如A*算法、Dijkstra算法等）来实现该功能。

要求如下：

1. 首先，定义海洋环境的数据结构，包括海域的尺寸、障碍物的位置、以及起点和终点的坐标。
2. 实现一个函数或方法，根据提供的海洋环境信息，计算并返回从起点到终点的最优路径。你需要保证算法的效率和优化路径的准确性。
3. 编写代码以模拟不同的海洋环境（至少3种），并利用你的程序为每种环境下的无人潜水器计算最优路径。
4. 为你的程序编写文档，说明如何运行你的代码，以及如何添加新的海洋环境信息进行路径规划。
5. （可选）提供一个简单的用户界面（UI）或可视化工具，以图形化方式显示海洋环境、无人潜水器的路径等信息。

请确保你的代码具有良好的注释和清晰的结构，以便于理解和修改。

Design and develop a program to simulate an autonomous path planning system for intelligent marine equipment. This system should determine the optimal path for an unmanned underwater vehicle (UUV) to travel from a starting point to a destination, based on given information about the marine environment. The environmental data may include, but is not limited to, the dimensions of the marine area, the presence of obstacles (such as coral reefs or deep-sea trenches), and the coordinates of the start and end points. You are expected to implement this functionality using an appropriate search or path planning algorithm, such as the A* algorithm or Dijkstra's algorithm.

Requirements:

1. Define a data structure to represent the marine environment, including the dimensions of the area, the positions of obstacles, and the coordinates of the start and end points.
2. Implement a function or method that takes the environmental data as input and computes the optimal path from the start to the destination. Your implementation should ensure both efficiency and path accuracy.
3. Write code to simulate at least three different marine environments. For each environment, use your program to calculate the optimal path for the UUV.
4. Create documentation for your program, explaining how to run the code and how to add new environmental data for path planning.
5. *(Optional)* Provide a simple user interface (UI) or visualization tool to graphically display the marine environment, the UUV's path, and other relevant information.

Please ensure your code is well-commented and clearly structured to facilitate understanding and future modifications.

Response

为了构建这个程序，我们将按照要求分步骤来执行。请注意，以下是一个简化的示例，旨在演示概念和基本的实现方法，并不涉及复杂的异常处理或高级优化技巧。

1. 定义海洋环境的数据结构

首先，我们需要定义海洋环境的数据结构。我们将创建一个简单的类`OceanEnvironment`，它包含海域的尺寸、障碍物的位置列表、起点和终点的坐标。
......

2. 实现路径规划函数

接下来，我们将使用A*算法来实现路径规划函数，因为A*算法比较适合在具有多种障碍物的环境中寻找最优路径。我们定义函数`find_path`来实现该算法。
......

3. 模拟不同的海洋环境并计算最优路径

现在，我们可以创建几个海洋环境的实例，并使用我们的路径规划函数来寻找最优路径。
......

4. 文档说明

为了运行上面的代码，你需要确保Python环境已安装。然后，将代码保存到一个`.py`文件中，在命令行或IDE中运行该文件即可看到不同环境下的最优路径结果。 若要添加新的海洋环境信息，仅需实例化`OceanEnvironment`类并按照实例的结构提供尺寸、障碍、起点和终点即可。

5. 用户界面(UI)

提供一个图形化界面或可视化工具超出了本说明文档的范畴，但是你可以利用像`matplotlib`这样的Python库来绘制海洋环境和找到的路径，从而为用户提供视觉反馈。

这样，我们就完成了一个简单的自动路径规划系统的设计与编写。

To build this program, we will follow a step-by-step approach as required. Please note that the following is a simplified example intended to demonstrate the core concepts and basic implementation. It does not include complex exception handling or advanced optimization techniques.

1. Define the Data Structure for the Ocean Environment

First, we need to define the data structure for the ocean environment. We'll create a simple class called `OceanEnvironment`, which includes the dimensions of the ocean area, a list of obstacle positions, and the coordinates of the start and end points.
......

2. Implement the Path Planning Function

Next, we'll implement the path planning function using the A* algorithm, as it is well-suited for finding optimal paths in environments with multiple obstacles. We'll define a function named `find_path` to carry out this algorithm.
......

3. Simulate Different Ocean Environments and Compute Optimal Paths

Now, we can create several instances of ocean environments and use our path planning function to find the optimal path for each case.
......

4. Documentation

To run the code above, make sure you have Python installed on your system. Then, save the code in a `.py` file and run it through the command line or an IDE to view the optimal path results in different environments. To add a new ocean environment, simply instantiate the `OceanEnvironment` class and provide the size, obstacles, start point, and end point according to the structure of the example.

5. User Interface (UI)

Creating a graphical interface or visualization tool is beyond the scope of this guide, but you can use Python libraries such as `matplotlib` to draw the ocean environment and the calculated path, providing visual feedback to the user.

With that, we've completed the design and development of a simple automated path planning system.

Figure 4: An example prompt and its corresponding response from the preference dataset constructed using TaP. Both the prompt and the response were generated by GPT-4. Portions of the response that were excessively long have been omitted for brevity and are indicated by blue ellipses.

Prompts for Generating Question Types

System Prompt	请根据指定的学科领域，尽可能详细地列举出相关的题型并附上每种题型的简介，题型和对应的简介由“：”连接。请注意，只需生成题型和简介，无需给出具体的题目示例。为了清晰地区分每一种题型及其简介，请在每条生成的题型及其简介的前后加上三引号（"""），具体的生成格式为"""题型：简介"""。 Please provide a comprehensive list of relevant question types for the specified subject, along with a brief description of each. Connect each question type and its corresponding description with a colon (":"). Only generate the question types and their descriptions; there is no need to provide specific question examples. To clearly distinguish each question type and its description, enclose each in triple quotation marks ("""). The specific format is as follows: """Question Type: Description""".
User Prompt of First Turn	学科：`{subject}` Subject: `{subject}`
User Prompt of Second Turn	请尽可能详细地列举出与`{subject}`相关的更多题型，并确保新生成的题型与已有的题型不重复。 Please provide a detailed list of additional question types related to `{subject}`, ensuring that each new type is distinct from the existing ones.
User Prompt of Third Turn	请尽可能详尽地继续列举出与`{subject}`相关的更多题型，并确保新生成的题型与已有的题型不重复。 Please continue providing a comprehensive list of additional question types related to `{subject}`, ensuring that each new type is distinct from the existing ones.

Figure 5: Prompts for generating different question types with brief descriptions of each type.

Prompts for Refining the Description of Question Types

System Prompt	请在给定的学科领域、题型和简介的基础上，对题型简介进行改写，使其表述更加规范、清晰，同时更贴近生活实际。请直接生成改写后的题型简介，无需生成其他内容。 Given the subject and the specified question type, along with its corresponding description, refine the description to ensure clarity, standardization, and alignment with real-life contexts. Provide the revised description directly without any additional content.

User Prompt of First Turn	### 学科领域
	`{subject}`
	
	### 题型
	`{question_type}`
	
	### 简介
	`{description}`
	
	### Subject
	`{subject}`
	
	### Question Type
	`{question_type}`
	
	### Description
	`{description}`

Figure 6: Prompts for refining the description of various question types.

Prompts for Generating Prompts

System Prompt

请根据所给的学科领域、题目类型以及简介，构思一个具体的指令。请确保生成的指令内容合理、完整、表述清晰，并且贴近生活实际，同时具有一定的难度。在生成指令时，请以“### 指令\n”作为开头。

Please craft a prompt based on the provided subject, question type, and description. The prompt should be realistic, comprehensive, and clearly articulated, with a moderate level of complexity. Additionally, ensure it is closely related to real-world scenarios. When presenting the prompt, begin with "### Prompt\n".

User Prompt of First Turn

```
### 学科领域
{subject}

### 题型
{question_type}

### 简介
{description}

### Subject
{subject}

### Question Type
{question_type}

### Description
{description}
```

User Prompt of Second Turn

请对上述生成的指令进行评估，判断是否因缺乏必要的输入而无法遵循，并输出评估结果（“是”或“否”）。如果输出的评估结果为“是”，请按照原有的格式重新生成上述指令，确保新生成的指令包含所有必要的输入。如果输出的评估结果为“否”，则无需再输出其他内容。

Please determine whether the generated prompt lacks necessary input, making it impossible to follow. Output either "Yes" or "No" as the evaluation result. If the result is "Yes," regenerate the prompt in its original format, ensuring that all required input is included. If the result is "No," output only "No" without any additional content.

Figure 7: Prompts for generating prompts based on specified question types and their corresponding refined descriptions.

Prompts for Checking Prompts

System Prompt

请你根据大语言模型的局限性，对给定指令的合理性进行评估。评估标准基于大语言模型是否具备执行该指令的能力。如果由于大语言模型的局限性导致其无法执行该指令，那么认为这个指令是不合理的。请你首先阐述评估理由，然后在最后一行给出评估结果（“合理”或“不合理”）。

大语言模型的局限性如下：
1. 无法直接与真实世界交互：大语言模型只能通过文本与用户交流，而无法直接操作物体、控制设备或影响物理世界。
2. 缺乏感官能力：大语言模型缺乏视觉、听觉、味觉、触觉、嗅觉等感官能力，因此无法感知或理解与这些感官相关的信息。
3. 无法处理非文本信息：大语言模型只能处理文本数据，无法理解或生成图像、音频、视频等非文本信息。
4. 无法访问实时信息：大语言模型的知识是静态的，它们无法实时获取和更新最新的新闻、事件或其他动态信息。

Please assess the reasonableness of a given instruction in light of the limitations of large language models. The criterion for evaluation is whether the model can successfully follow the instruction. If the model's inherent limitations prevent it from doing so, the instruction should be deemed unreasonable. Begin by explaining the rationale behind your evaluation, then state the final evaluation result ("Reasonable" or "Unreasonable") on the last line.

The key limitations of large language models are as follows:
1. Inability to directly interact with the real world: Large language models can only communicate via text and cannot manipulate objects, control devices, or influence the physical world directly.
2. Lack of sensory capabilities: These models do not possess sensory faculties such as vision, hearing, taste, touch, or smell, making them unable to perceive or interpret information from these senses.
3. Inability to process non-text information: Large language models can only handle textual data and are incapable of understanding or generating images, audio, video, or other non-textual content.
4. Inability to access real-time information: The knowledge base of large language models is static, meaning they cannot retrieve or update real-time news, events, or other dynamic information.

User Prompt of First Turn

指令："""{instruction}"""

Instruction: """{instruction}"""

Figure 8: Prompts for evaluating the reasonableness of a given instruction, considering the inherent limitations of LLMs.

Prompts for Generating Responses

System Prompt

你是一个出色的人工智能助手，你的回复必须与用户的指令紧密相关，同时确保所提供的信息准确无误。此外，你的回复必须表达清晰，内容完整。然而，请注意，你存在以下的限制：
1. 无法直接与真实世界交互：你只能通过文本与用户交流，而无法直接操作物体、控制设备或影响物理世界。
2. 缺乏感官能力：你缺乏视觉、听觉、味觉、触觉、嗅觉等感官能力，因此无法感知或理解与这些感官相关的信息。
3. 无法处理非文本信息：你只能处理文本数据，无法理解或生成图像、音频、视频等非文本信息。
4. 无法访问实时信息：你的知识是静态的，你无法实时获取和更新最新的新闻、事件或其他动态信息。
当你因为以上限制而无法执行用户指令时，请向用户明确说明你无法执行他们的指令，并详细解释具体的原因。

You are an exceptional AI assistant, and your responses must strictly adhere to user instructions while ensuring the accuracy and reliability of the provided information. Additionally, your responses should be clear and comprehensive. However, you must acknowledge the following inherent limitations:
1. Inability to interact directly with the real world: You can only communicate with users through text and cannot physically manipulate objects, control devices, or influence the physical world.
2. Lack of sensory capabilities: You do not possess sensory faculties such as vision, hearing, taste, touch, or smell, making them unable to perceive or interpret information from these senses.
3. Inability to process non-text information: You can only handle textual data and are incapable of understanding or generating images, audio, video, or other non-textual content.
4. Inability to access real-time information: Your knowledge is static, and you cannot retrieve or update the latest news, events, or other dynamic information in real time.
If a user's request falls outside your capabilities due to these limitations, you should clearly illustrate your inability to fulfill the request and provide a detailed explanation of the reasons.

Figure 9: Prompts used for response generation.

Prompts for Scoring Responses

System Prompt

你的任务是根据提供的评分标准，对两个大语言模型生成的回复进行评分。以下是一个指令和对应的两个大语言模型的回复。请你分别从相关性、准确性、清晰性和完整性四个评估维度对两个大语言模型的回复进行评分。评分范围为1至10分，1分为最低，10分为最高。在评分时请确保两个大语言模型的回复的顺序不会影响你的评分结果。

各评估维度的含义如下：

- **相关性**：评估模型生成的回复是否与用户输入紧密相关，且能提供有价值的信息。
- **准确性**：评估模型生成的回复是否准确无误，没有事实错误或误导性信息，能够有效解决用户的问题或满足用户的需求。
- **清晰性**：评估模型生成的回复在语法、拼写和标点使用上的正确性，表达是否清晰、易于理解，没有含糊不清之处。
- **完整性**：评估模型生成的回复是否提供了足够且完整的信息，能够充分回答用户的问题或满足用户的需求，而无需进一步解释或补充。

请先详细说明评分依据，然后在最后给出具体的分数，并严格按照以下格式输出具体的分数：

\### 大语言模型1的回复评分
- 相关性：[[Rating]]
- 准确性：[[Rating]]
- 清晰性：[[Rating]]
- 完整性：[[Rating]]

\### 大语言模型2的回复评分
- 相关性：[[Rating]]
- 准确性：[[Rating]]
- 清晰性：[[Rating]]
- 完整性：[[Rating]]

Your task is to evaluate the responses generated by two large language models based on the provided evaluation criteria. Below is an instruction followed by responses from the two models. Please assess each response according to four dimensions: relevance, correctness, clarity, and completeness. The rating scale ranges from 1 to 10, with 1 being the lowest and 10 being the highest. Ensure that the order of the responses does not influence your evaluation.

The definitions of each evaluation dimension are as follows:

- **Relevance**: Assess whether the model's response is closely related to the user's input and provides valuable information.
- **Correctness**: Evaluate whether the model's response is accurate, free from factual errors or misleading information, and effectively addresses the user's question or meets their needs.
- **Clarity**: Examine the model's response for proper grammar, spelling, and punctuation, and determine whether the expression is clear, easy to understand, and free from ambiguity.
- **Completeness**: Determine whether the model's response provides sufficient and comprehensive information, fully addressing the user's question or meeting their needs without requiring further clarification or supplementation.

Please first provide a detailed explanation of your evaluation criteria and rationale. Then, present the scores strictly following the format below:

\### Scores for Response from Large Language Model 1
- Relevance: [[Rating]]
- correctness: [[Rating]]
- Clarity: [[Rating]]
- Completeness: [[Rating]]

\### Scores for Response from Large Language Model 2
- Relevance: [[Rating]]
- correctness: [[Rating]]
- Clarity: [[Rating]]
- Completeness: [[Rating]]

User Prompt of First Turn

\### 指令
{instruction}

\### 大语言模型1的回复
{response_1}

\### 大语言模型2的回复
{response_2}

\### Instruction
{instruction}

\### Response from Large Language Model 1
{response_1}

\### Response from Large Language Model 2
{response_2}

Figure 10: Prompts for conducting pairwise comparisons, where two responses are evaluated on a scale from 1 to 10 across four evaluation dimensions: (1) relevance, (2) correctness, (3) clarity, and (4) completeness.

Instruction and Corresponding GPT-4 Assessment

Instruction

请利用GIS软件或其他地理空间信息工具完成以下任务：

1. **地形分析**：
   - 首先，导入提供的数字高程模型（DEM）数据。
   - 使用该DEM数据，生成研究区域的坡度图和高程图。确保图中清晰地标注出高度和坡度的等级分布。

2. **水文分析**：
   - 基于DEM数据，利用水文分析工具，确定并绘制研究区域内的主要流域边界和河流网络。
   - 计算主要河流的流长和流域的面积。

3. **空间插值**：
   - 假定你有来自10个不同地点的温度观测数据（地点坐标及其对应温度值将提供）。
   - 应用空间插值技术，创建一个温度分布图，覆盖整个研究区域。

4. **结果展示与分析**：
   - 对上述生成的坡度图、高程图、流域边界图及河流网络图、温度分布图进行合理的图层叠加，分析地形影响下的温度分布特征。
   - 准备一份报告，总结你在地形分析、水文分析和空间插值过程中所使用的方法、步骤和技巧，包括遇到的挑战以及如何克服它们。

提供的数据包括数字高程模型（DEM）数据和10个地点的温度观测数据。所有分析步骤必须在GIS软件或其他相应的地理空间工具中完成。请确保你的分析结果有足够的解释和详细的步骤记录，以供评审。

Please use GIS software or other geospatial tools to complete the following tasks:

1. **Terrain Analysis**:
   - First, import the provided Digital Elevation Model (DEM) data.
   - Using the DEM data, generate a slope map and an elevation map of the study area. Make sure the maps clearly indicate the distribution of elevation and slope levels.

2. **Hydrological Analysis**:
   - Based on the DEM data, use hydrological analysis tools to identify and map the main watershed boundaries and river networks within the study area.
   - Calculate the flow length of the main rivers and the area of the watershed.

3. **Spatial Interpolation**:
   - Assume you have temperature observations from 10 different locations (coordinates and corresponding temperature values will be provided).
   - Apply spatial interpolation techniques to create a temperature distribution map covering the entire study area.

4. **Results Visualization and Analysis**:
   - Overlay the slope map, elevation map, watershed boundaries, river network, and temperature distribution map to analyze how topography influences temperature distribution.
   - Prepare a report summarizing the methods, steps, and techniques used in the terrain analysis, hydrological analysis, and spatial interpolation. The report should also describe any challenges encountered and how they were addressed.

The provided data include the DEM and temperature observations from 10 locations. All analysis must be conducted using GIS software or other appropriate geospatial tools. Be sure to include detailed documentation and sufficient explanations of your results for evaluation purposes.

GPT-4 Assessment

评估理由：该指令要求使用GIS软件或其他地理空间信息工具来完成一系列复杂的地理空间分析任务，包括地形分析、水文分析、空间插值以及结果的展示与分析。大语言模型无法直接操作GIS软件或其他地理空间信息工具来导入数据、生成图表、执行空间分析或创建报告。此外，这些任务通常需要图形界面的交互、实时数据处理和分析，以及对地理空间数据的直接操作，这些都超出了大语言模型的能力范围。因此，根据大语言模型的局限性，它无法完成这一系列要求的操作。

评估结果：不合理。

Evaluation Rationale: The instruction requires the use of GIS software or other geospatial tools to carry out a series of complex spatial analysis tasks, including terrain analysis, hydrological analysis, spatial interpolation, and the visualization and interpretation of results. Large language models are not capable of directly operating GIS software or other geospatial tools to import data, generate maps, perform spatial analysis, or create reports. These tasks typically involve interactions through graphical user interfaces, real-time data processing and analysis, and direct manipulation of geospatial datasets—all of which are beyond the capabilities of a large language model. Therefore, due to these inherent limitations, the model is unable to complete the required operations.

Evaluation Result: Unreasonable.

Figure 11: A representative example from the sampled set in which GPT-4 correctly recognizes the instruction as exceeding the capabilities of LLMs. The figure includes both the instruction and GPT-4's corresponding assessment.

Model Name	#Parameters	Model Type
gpt-4-0125-preview	N/A	N/A
DeepSeek-V2	236B (21B)	MoE
Baichuan-13B-Chat	13B	Dense
Baichuan2-7B-Chat	7B	Dense
Baichuan2-13B-Chat	13B	Dense
Qwen1.5-0.5B-Chat	0.5B	Dense
Qwen1.5-1.8B-Chat	1.8B	Dense
Qwen1.5-4B-Chat	4B	Dense
Qwen1.5-7B-Chat	7B	Dense
Qwen1.5-14B-Chat	14B	Dense
Qwen1.5-32B-Chat	32B	Dense
Qwen1.5-72B-Chat	72B	Dense
Qwen1.5-110B-Chat	110B	Dense
Qwen1.5-MoE-A2.7B-Chat	14B (3B)	MoE
Qwen2-0.5B-Instruct	0.5B	Dense
Qwen2-1.5B-Instruct	1.5B	Dense
Qwen2-7B-Instruct	7B	Dense
Qwen2-72B-Instruct	72B	Dense
Qwen2-57B-A14B-Instruct	57B (14B)	MoE
Yi-6B-Chat	6B	Dense
Yi-34B-Chat	34B	Dense
Yi-1.5-6B-Chat	6B	Dense
Yi-1.5-9B-Chat	9B	Dense
Yi-1.5-34B-Chat	34B	Dense
Llama-3-8B-Instruct	8B	Dense
Llama-3-70B-Instruct	70B	Dense
Llama-3.1-8B-Instruct	8B	Dense
Llama-3.1-70B-Instruct	70B	Dense
gemma-2b-it	2B	Dense
gemma-7b-it	7B	Dense
gemma-1.1-2b-it	2B	Dense
gemma-1.1-7b-it	7B	Dense
gemma-2-2b-it	2B	Dense
gemma-2-9b-it	9B	Dense
gemma-2-27b-it	27B	Dense
Mistral-7B-Instruct-v0.3	7B	Dense
Mistral-Large-Instruct-2407	123B	Dense
Mistral-Nemo-Instruct-2407	12B	Dense
Mixtral-8x22B-Instruct-v0.1	141B (39B)	MoE
Mixtral-8x7B-Instruct-v0.1	47B (13B)	MoE
c4ai-command-r-v01	35B	Dense
c4ai-command-r-plus	104B	Dense
aya-23-8B	8B	Dense
aya-23-35B	35B	Dense
chatglm3-6b	6B	Dense
glm-4-9b-chat	9B	Dense
AquilaChat-7B	7B	Dense
AquilaChat2-7B	7B	Dense
AquilaChat2-34B	34B	Dense
dbrx-instruct	132B (36B)	MoE
internlm-chat-7b	7B	Dense
internlm-chat-20b	20B	Dense
internlm2-chat-1_8b	1.8B	Dense
internlm2-chat-7b	7B	Dense
internlm2-chat-20b	20B	Dense
internlm2_5-1_8b-chat	1.8B	Dense
internlm2_5-7b-chat	7B	Dense
internlm2_5-20b-chat	20B	Dense
MiniCPM-2B-dpo-bf16	2B	Dense
MiniCPM-2B-sft-bf16	2B	Dense
MiniCPM-1B-sft-bf16	1B	Dense
Orion-14B-Chat	14B	Dense
XVERSE-7B-Chat	7B	Dense
XVERSE-13B-Chat	13B	Dense
XVERSE-65B-Chat	65B	Dense

Table 5: Model architectures and parameter counts of LLMs employed in response generation. Parameter counts enclosed in parentheses represent the number of activated parameters for MoE LLMs.

Model	Dataset	AlignBench						MT-Bench-zh	
		GPT-4o			DeepSeek-V3				
		Reasoning	Language	Overall	Reasoning	Language	Overall	GPT-4o	DeepSeek-V3
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.13	4.88	4.51	4.44	5.36	4.90	5.25	5.20
	Chinese-DPO-Pairs	5.12	5.98	5.55	4.53	5.52	5.02	**5.99**	**5.71**
	TaP (GPT-4)	**5.20**	**6.11**	**5.66**	**4.71**	**5.62**	**5.17**	5.80	5.68
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	4.61	5.27	4.94	4.62	5.55	5.08	5.86	5.76
	Chinese-DPO-Pairs	4.88	5.51	5.20	**5.07**	**5.77**	**5.42**	**6.08**	5.79
	TaP (GPT-4)	**5.69**	**6.27**	**5.98**	4.98	5.68	5.33	6.00	**5.81**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.71	5.56	5.13	4.46	5.43	4.95	5.21	5.19
	Chinese-DPO-Pairs	4.41	5.33	4.87	4.31	**5.52**	4.91	5.23	5.14
	TaP (GPT-4)	**4.94**	**5.81**	**5.38**	**4.64**	5.41	**5.03**	**5.39**	**5.49**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.24	6.44	6.34	5.67	5.98	5.82	6.25	6.19
	Chinese-DPO-Pairs	5.79	6.02	5.90	**5.89**	6.04	**5.97**	6.35	6.31
	TaP (GPT-4)	**6.29**	**6.62**	**6.46**	5.73	**6.15**	5.94	**6.36**	**6.38**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.72	5.27	4.50	**3.41**	4.97	4.19	4.93	4.89
	Chinese-DPO-Pairs	2.79	4.52	3.66	3.10	5.03	4.06	4.75	4.93
	TaP (GPT-4)	**3.78**	**5.57**	**4.68**	3.33	**5.28**	**4.30**	**5.14**	**4.94**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.94	5.25	4.60	**3.61**	4.86	4.24	5.25	5.04
	Chinese-DPO-Pairs	3.72	5.09	4.40	3.30	4.89	4.09	5.21	5.04
	TaP (GPT-4)	**4.04**	**5.59**	**4.81**	3.43	**4.94**	**4.18**	**5.46**	**5.28**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.75	4.75	4.25	3.81	5.14	4.47	4.87	4.85
	Chinese-DPO-Pairs	4.05	5.48	4.77	3.73	5.25	4.49	5.03	4.99
	TaP (GPT-4)	**4.49**	**5.89**	**5.19**	**3.96**	**5.55**	**4.75**	**5.52**	**5.50**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.85	5.20	4.53	3.56	4.96	4.26	5.27	5.11
	Chinese-DPO-Pairs	4.14	5.32	4.73	3.55	4.91	4.23	5.45	5.18
	TaP (GPT-4)	**4.42**	**5.58**	**5.00**	**3.91**	**5.23**	**4.57**	**5.63**	**5.26**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	5.61	5.90	5.75	5.69	5.93	5.81	5.78	5.78
	Chinese-DPO-Pairs	6.05	6.30	6.18	5.85	**6.17**	6.01	5.86	5.78
	TaP (GPT-4)	**6.53**	**6.49**	**6.51**	**6.02**	6.16	**6.09**	**6.32**	**6.21**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.61	6.50	6.55	6.42	6.34	6.38	6.41	**6.51**
	Chinese-DPO-Pairs	6.37	6.40	6.38	6.48	**6.59**	6.53	**6.83**	6.46
	TaP (GPT-4)	**7.42**	**7.12**	**7.27**	**6.84**	6.50	**6.67**	6.58	6.43

Table 6: Performance comparison of LLMs trained with PPO using different datasets on AlignBench and MT-Bench-zh. The model names include two possible suffixes: “Open” and “TaP.” The “Open” suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas “TaP” denotes initialization from models trained on a dataset constructed by TaP. Additionally, “GPT-4o” and “DeepSeek-V3” specify that responses were evaluated using GPT-4o and DeepSeek-V3, respectively.

Model	Dataset	GPT-4o			DeepSeek-V3		
		First turn	Second turn	Average	First turn	Second turn	Average
Qwen2.5-3B	Firefly	5.73	3.10	4.41	5.88	3.03	4.45
	Alpaca-GPT-4-ZH (Peng et al., 2023)	6.29	3.36	4.83	6.39	2.98	4.68
	COIG (Zhang et al., 2023a)	4.18	2.53	3.35	4.56	2.31	3.44
	MOSS-SFT (Sun et al., 2024)	6.31	3.00	4.66	6.01	2.74	4.38
	COIG-CQIA (Bai et al., 2024)	5.99	2.95	4.47	6.01	2.69	4.35
	Infinity-Instruct	6.61	3.18	4.89	6.46	2.90	4.68
	BELLE-SFT (BELLEGroup, 2023)	6.11	3.18	4.65	6.04	2.94	4.49
	TaP-SFT (GPT-4)	**7.63**	**3.96**	**5.79**	7.36	**3.69**	5.53
	TaP-SFT (DeepSeek-V2)	7.56	3.78	5.67	**7.59**	3.61	**5.60**
Qwen2.5-7B	Firefly	5.75	3.10	4.43	6.09	2.85	4.47
	Alpaca-GPT-4-ZH (Peng et al., 2023)	6.64	3.40	5.02	6.54	3.24	4.89
	COIG (Zhang et al., 2023a)	4.40	2.55	3.48	4.86	2.46	3.66
	MOSS-SFT (Sun et al., 2024)	6.94	3.18	5.06	6.95	2.85	4.90
	COIG-CQIA (Bai et al., 2024)	6.50	2.96	4.73	6.49	2.54	4.51
	Infinity-Instruct	6.61	3.31	4.96	6.96	2.94	4.95
	BELLE-SFT (BELLEGroup, 2023)	6.36	3.50	4.93	6.44	3.23	4.83
	TaP-SFT (GPT-4)	8.01	**4.30**	**6.16**	8.16	**3.81**	**5.99**
	TaP-SFT (DeepSeek-V2)	**8.39**	3.81	6.10	**8.49**	3.45	5.97
Llama-3.1-8B	Firefly	4.79	2.65	3.72	4.94	2.44	3.69
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.46	2.95	4.21	5.66	2.90	4.28
	COIG (Zhang et al., 2023a)	3.39	1.98	2.68	3.63	1.86	2.74
	MOSS-SFT (Sun et al., 2024)	5.28	3.15	4.21	5.35	2.76	4.06
	COIG-CQIA (Bai et al., 2024)	4.36	2.26	3.32	4.66	2.03	3.34
	Infinity-Instruct	6.06	3.35	4.71	6.05	3.15	4.60
	BELLE-SFT (BELLEGroup, 2023)	5.98	3.27	4.63	6.30	3.18	4.74
	TaP-SFT (GPT-4)	**6.80**	3.69	**5.24**	**6.83**	3.08	**4.95**
	TaP-SFT (DeepSeek-V2)	6.53	**3.70**	5.11	6.51	**3.26**	4.89
Gemma-2-9B	Firefly	5.00	2.84	3.92	4.98	2.56	3.77
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.56	3.21	4.39	5.51	2.66	4.09
	COIG (Zhang et al., 2023a)	3.35	2.03	2.69	3.49	1.81	2.65
	MOSS-SFT (Sun et al., 2024)	5.31	2.98	4.14	5.44	3.03	4.23
	COIG-CQIA (Bai et al., 2024)	4.26	2.46	3.36	4.19	2.08	3.13
	Infinity-Instruct	6.34	3.23	4.78	6.08	2.83	4.45
	BELLE-SFT (BELLEGroup, 2023)	6.13	3.25	4.69	6.46	**3.28**	4.87
	TaP-SFT (GPT-4)	**7.03**	**3.71**	**5.37**	**6.89**	3.16	**5.03**
	TaP-SFT (DeepSeek-V2)	6.50	3.30	4.90	6.46	2.93	4.69
Qwen2.5-14B	Firefly	6.11	3.43	4.77	6.36	3.23	4.79
	Alpaca-GPT-4-ZH (Peng et al., 2023)	7.14	3.75	5.44	7.01	3.71	5.36
	COIG (Zhang et al., 2023a)	4.64	2.63	3.63	4.69	2.43	3.56
	MOSS-SFT (Sun et al., 2024)	6.99	3.53	5.26	7.00	3.08	5.04
	COIG-CQIA (Bai et al., 2024)	6.75	3.09	4.92	6.74	2.83	4.78
	Infinity-Instruct	7.00	3.66	5.33	6.93	3.19	5.06
	BELLE-SFT (BELLEGroup, 2023)	6.60	3.39	4.99	6.65	3.00	4.83
	TaP-SFT (GPT-4)	**8.29**	**4.40**	**6.34**	**8.31**	4.04	6.18
	TaP-SFT (DeepSeek-V2)	8.20	4.04	6.12	8.29	**4.09**	**6.19**

Table 7: Performance comparison of five LLMs on MT-Bench-zh after **supervised fine-tuning** with various datasets. “GPT-4o” and “DeepSeek-V3” indicate that the responses were evaluated using GPT-4o and DeepSeek-V3, respectively.

Model	Dataset	GPT-4o			DeepSeek-V3		
		First turn	Second turn	Average	First turn	Second turn	Average
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.71	3.35	5.03	6.79	3.09	4.94
	Chinese-DPO-Pairs	6.53	3.56	5.04	6.66	3.24	4.95
	TaP (GPT-4)	**7.69**	**4.08**	**5.88**	**7.76**	**3.49**	**5.63**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	7.46	3.98	5.72	7.18	3.31	5.24
	Chinese-DPO-Pairs	7.63	**4.19**	5.91	7.68	**3.69**	5.68
	TaP (GPT-4)	**7.94**	4.13	**6.03**	**7.98**	3.60	**5.79**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	7.09	3.41	5.25	6.96	3.00	4.98
	Chinese-DPO-Pairs	7.09	3.36	5.23	7.26	**3.20**	5.23
	TaP (GPT-4)	**7.54**	**3.55**	**5.54**	**7.59**	3.18	**5.38**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	8.10	4.10	6.10	7.99	3.71	5.85
	Chinese-DPO-Pairs	8.04	4.29	6.16	**8.24**	3.73	5.98
	TaP (GPT-4)	**8.15**	**4.58**	**6.36**	8.23	**4.00**	**6.11**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.34	3.36	4.85	6.38	2.78	4.58
	Chinese-DPO-Pairs	6.51	3.38	4.94	6.50	**3.25**	4.88
	TaP (GPT-4)	**7.11**	**3.70**	**5.41**	**7.11**	3.16	**5.14**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.81	3.56	5.19	6.59	3.09	4.84
	Chinese-DPO-Pairs	6.85	3.75	5.30	6.69	2.99	4.84
	TaP (GPT-4)	**6.99**	**4.20**	**5.59**	**7.00**	**3.31**	**5.16**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.21	3.33	4.77	6.14	3.10	4.62
	Chinese-DPO-Pairs	6.43	3.40	4.91	6.65	3.04	4.84
	TaP (GPT-4)	**7.29**	**3.79**	**5.54**	**7.29**	**3.18**	**5.23**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.95	3.84	5.39	6.81	3.21	5.01
	Chinese-DPO-Pairs	7.00	4.09	5.54	7.00	**3.40**	5.20
	TaP (GPT-4)	**7.44**	**4.13**	**5.78**	**7.49**	3.16	**5.33**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	7.16	3.66	5.41	7.50	3.46	5.48
	Chinese-DPO-Pairs	7.36	3.71	5.54	7.64	3.53	5.58
	TaP (GPT-4)	**7.64**	**4.05**	**5.84**	**7.81**	**3.96**	**5.89**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	8.21	4.29	6.25	8.30	3.93	6.11
	Chinese-DPO-Pairs	8.31	4.38	6.34	8.33	3.95	6.14
	TaP (GPT-4)	**8.50**	**4.83**	**6.66**	**8.54**	**4.16**	**6.35**

Table 8: Performance comparison of LLMs trained with **DPO** using different datasets on MT-Bench-zh. The model names include two possible suffixes: “Open” and “TaP.” The “Open” suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas “TaP” denotes initialization from models trained on a dataset constructed by TaP. Additionally, “GPT-4o” and “DeepSeek-V3” specify that responses were evaluated using GPT-4o and DeepSeek-V3, respectively.

Model	Dataset	GPT-4o			DeepSeek-V3		
		First turn	Second turn	Average	First turn	Second turn	Average
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.86	3.64	5.25	6.91	3.49	5.20
	Chinese-DPO-Pairs	**7.86**	**4.13**	**5.99**	**7.78**	3.65	**5.71**
	TaP (GPT-4)	7.84	3.76	5.80	7.70	**3.66**	5.68
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	7.51	4.21	5.86	7.65	**3.86**	5.76
	Chinese-DPO-Pairs	7.55	**4.60**	**6.08**	7.79	3.80	5.79
	TaP (GPT-4)	**7.80**	4.20	6.00	**7.99**	3.64	**5.81**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	7.05	3.36	5.21	7.21	3.18	5.19
	Chinese-DPO-Pairs	7.10	3.36	5.23	7.03	3.25	5.14
	TaP (GPT-4)	**7.31**	**3.46**	**5.39**	**7.56**	**3.41**	**5.49**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	8.15	**4.35**	6.25	8.35	4.03	6.19
	Chinese-DPO-Pairs	8.36	4.34	6.35	8.45	4.18	6.31
	TaP (GPT-4)	**8.54**	4.19	**6.36**	**8.55**	**4.21**	**6.38**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.34	**3.53**	4.93	6.46	**3.33**	4.89
	Chinese-DPO-Pairs	6.25	3.24	4.75	6.64	3.21	4.93
	TaP (GPT-4)	**6.88**	3.41	**5.14**	**6.79**	3.10	**4.94**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.61	3.89	5.25	6.85	3.24	5.04
	Chinese-DPO-Pairs	6.79	3.63	5.21	**7.01**	3.08	5.04
	TaP (GPT-4)	**6.94**	**3.99**	**5.46**	6.99	**3.58**	**5.28**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.41	3.33	4.87	6.49	3.21	4.85
	Chinese-DPO-Pairs	6.58	3.48	5.03	6.64	3.34	4.99
	TaP (GPT-4)	**7.49**	**3.55**	**5.52**	**7.53**	**3.48**	**5.50**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.85	3.69	5.27	7.05	3.16	5.11
	Chinese-DPO-Pairs	6.93	3.98	5.45	7.04	**3.33**	5.18
	TaP (GPT-4)	**7.25**	**4.01**	**5.63**	**7.29**	3.23	**5.26**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	7.59	3.98	5.78	7.80	3.76	5.78
	Chinese-DPO-Pairs	7.80	3.93	5.86	7.81	3.74	5.78
	TaP (GPT-4)	**8.19**	**4.45**	**6.32**	**8.29**	**4.14**	**6.21**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	8.29	4.54	6.41	8.60	**4.43**	**6.51**
	Chinese-DPO-Pairs	**8.86**	**4.80**	**6.83**	8.75	4.16	6.46
	TaP (GPT-4)	8.50	4.66	6.58	**8.83**	4.04	6.43

Table 9: Performance comparison of LLMs trained with **PPO** using different datasets on MT-Bench-zh. The model names include two possible suffixes: “Open” and “TaP.” The “Open” suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas “TaP” denotes initialization from models trained on a dataset constructed by TaP. Additionally, “GPT-4o” and “DeepSeek-V3” specify that responses were evaluated using GPT-4o and DeepSeek-V3, respectively.

Model	Dataset	Reasoning			Language							Overall
		Math.	Logi.	Avg.	Pro.	Chi.	Fund.	Writ.	Open.	Role.	Avg.	
Qwen2.5-3B	Firefly	3.59	2.86	3.22	4.56	4.60	4.07	4.80	3.79	4.19	4.34	3.78
	Alpaca-GPT-4-ZH (Peng et al., 2023)	4.67	3.65	4.16	4.86	4.61	4.84	4.77	5.42	4.69	4.87	4.51
	COIG (Zhang et al., 2023a)	2.96	2.49	2.73	4.02	4.44	3.68	3.84	4.61	4.16	4.12	3.42
	MOSS-SFT (Sun et al., 2024)	4.30	3.34	3.82	4.96	4.60	4.99	4.99	5.11	5.09	4.96	4.39
	COIG-CQIA (Bai et al., 2024)	3.79	3.42	3.61	4.15	4.28	4.51	4.57	4.21	4.56	4.38	3.99
	Infinity-Instruct	4.56	3.48	4.02	5.10	4.52	4.74	5.44	5.26	5.63	5.11	4.57
	BELLE-SFT (BELLEGroup, 2023)	4.60	3.16	3.88	5.00	4.76	4.97	5.28	5.34	5.28	5.11	4.49
	TGPDG-SFT (GPT-4)	5.92	4.05	4.99	**5.59**	**5.50**	**5.26**	**6.28**	**6.39**	**6.07**	**5.85**	5.42
	TGPDG-SFT (DeepSeek-V2)	**6.03**	**4.48**	**5.25**	5.48	5.43	5.25	5.92	6.13	5.70	5.65	**5.45**
Qwen2.5-7B	Firefly	3.66	3.24	3.45	4.72	4.31	4.97	5.07	3.84	4.78	4.61	4.03
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.49	4.30	4.90	5.50	5.21	5.44	5.24	5.13	5.34	5.31	5.10
	COIG (Zhang et al., 2023a)	3.04	2.61	2.82	4.65	4.68	4.10	4.01	4.55	4.45	4.41	3.62
	MOSS-SFT (Sun et al., 2024)	5.10	4.32	4.71	5.46	4.78	4.87	5.35	5.68	5.41	5.26	4.98
	COIG-CQIA (Bai et al., 2024)	5.26	4.17	4.72	5.10	5.35	5.07	5.15	5.13	4.96	5.13	4.92
	Infinity-Instruct	5.44	4.40	4.92	5.37	5.16	5.69	5.89	5.74	5.96	5.63	5.28
	BELLE-SFT (BELLEGroup, 2023)	5.07	3.57	4.32	5.37	4.91	5.56	5.49	5.68	5.61	5.44	4.88
	TGPDG-SFT (GPT-4)	**7.42**	**5.83**	**6.62**	**6.37**	5.78	**5.93**	**6.41**	**6.82**	**6.30**	**6.27**	**6.45**
	TGPDG-SFT (DeepSeek-V2)	7.08	5.41	6.25	5.57	**5.95**	5.76	6.28	6.76	6.10	6.07	6.16
Llama-3.1-8B	Firefly	2.41	2.38	2.40	3.87	3.86	3.97	4.53	3.53	4.09	3.97	3.19
	Alpaca-GPT-4-ZH (Peng et al., 2023)	2.50	3.04	2.77	4.12	3.84	4.09	4.67	4.37	4.59	4.28	3.53
	COIG (Zhang et al., 2023a)	1.73	1.85	1.79	2.96	2.53	2.90	2.79	3.32	3.37	2.98	2.38
	MOSS-SFT (Sun et al., 2024)	2.63	3.05	2.84	4.23	3.66	4.34	4.62	4.97	4.66	4.41	3.63
	COIG-CQIA (Bai et al., 2024)	2.30	2.78	2.54	3.45	3.28	3.41	3.49	3.95	3.86	3.57	3.06
	Infinity-Instruct	3.23	3.07	3.15	4.76	4.21	4.79	**5.68**	5.42	5.57	5.07	4.11
	BELLE-SFT (BELLEGroup, 2023)	3.57	3.38	3.47	**4.92**	**4.24**	4.81	5.31	5.55	5.48	5.05	4.26
	TGPDG-SFT (GPT-4)	**3.83**	**3.90**	**3.87**	4.90	3.83	**4.85**	5.52	**6.05**	**5.71**	**5.14**	**4.50**
	TGPDG-SFT (DeepSeek-V2)	3.48	3.73	3.61	3.98	3.64	4.60	5.41	5.58	5.28	4.75	4.18
Gemma-2-9B	Firefly	2.21	2.50	2.36	3.69	3.69	4.19	4.61	3.32	4.21	3.95	3.15
	Alpaca-GPT-4-ZH (Peng et al., 2023)	3.18	3.16	3.17	4.32	3.36	4.18	4.40	4.47	4.59	4.22	3.70
	COIG (Zhang et al., 2023a)	1.47	1.92	1.70	2.94	2.38	2.40	2.56	3.26	3.30	2.81	2.25
	MOSS-SFT (Sun et al., 2024)	2.72	2.96	2.84	4.00	3.38	3.89	4.55	5.16	4.30	4.21	3.53
	COIG-CQIA (Bai et al., 2024)	2.55	2.83	2.69	3.25	3.03	3.76	3.53	3.87	4.05	3.58	3.14
	Infinity-Instruct	3.92	3.52	3.72	5.06	4.19	5.12	**5.47**	5.61	5.83	**5.21**	4.47
	BELLE-SFT (BELLEGroup, 2023)	3.97	3.47	3.72	**5.11**	**4.33**	**5.26**	5.36	5.53	5.34	5.16	4.44
	TGPDG-SFT (GPT-4)	**4.50**	**3.71**	**4.10**	5.07	3.84	5.00	5.41	**5.79**	**5.91**	5.17	**4.64**
	TGPDG-SFT (DeepSeek-V2)	4.26	3.66	3.96	4.52	4.00	4.87	5.37	5.37	5.25	4.90	4.43
Qwen2.5-14B	Firefly	4.21	3.66	3.94	4.87	5.33	5.49	4.97	4.18	4.82	4.94	4.44
	Alpaca-GPT-4-ZH (Peng et al., 2023)	6.14	5.13	5.64	5.94	5.53	6.12	5.37	5.84	5.59	5.73	5.68
	COIG (Zhang et al., 2023a)	3.21	3.39	3.30	4.92	5.04	4.94	4.43	5.18	4.55	4.84	4.07
	MOSS-SFT (Sun et al., 2024)	5.30	5.20	5.25	5.57	5.05	5.75	5.75	5.74	5.66	5.59	5.42
	COIG-CQIA (Bai et al., 2024)	5.55	4.73	5.14	5.44	5.36	5.52	5.25	4.97	5.04	5.27	5.20
	Infinity-Instruct	5.96	4.55	5.26	5.86	5.37	6.03	6.19	6.03	6.16	5.94	5.60
	BELLE-SFT (BELLEGroup, 2023)	4.93	4.47	4.70	5.51	5.33	6.12	5.59	5.95	5.72	5.70	5.20
	TGPDG-SFT (GPT-4)	**7.50**	**6.02**	**6.76**	**6.87**	**5.76**	**6.54**	**6.65**	**6.71**	**6.53**	**6.51**	**6.64**
	TGPDG-SFT (DeepSeek-V2)	7.21	5.47	6.34	5.99	5.57	6.07	5.87	6.16	5.85	5.92	6.13

Table 10: Performance comparison of five LLMs on Alignbench after supervised fine-tuning with various datasets. The evaluation is conducted using **GPT-4o**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.

Model	Dataset	Reasoning			Language							Overall
		Math.	Logi.	Avg.	Pro.	Chi.	Fund.	Writ.	Open.	Role.	Avg.	
Qwen2.5-3B	Firefly	3.84	2.92	3.38	4.31	4.50	4.21	4.61	4.08	4.49	4.37	3.87
	Alpaca-GPT-4-ZH (Peng et al., 2023)	4.53	3.45	3.99	4.54	4.21	4.70	4.77	5.24	5.03	4.75	4.37
	COIG (Zhang et al., 2023a)	2.87	2.80	2.84	3.82	3.84	3.76	3.65	4.79	4.30	4.03	3.43
	MOSS-SFT (Sun et al., 2024)	4.24	3.22	3.73	4.78	4.24	4.66	5.00	5.13	5.09	4.82	4.27
	COIG-CQIA (Bai et al., 2024)	3.81	3.49	3.65	3.99	4.12	4.43	4.41	4.42	4.51	4.31	3.98
	Infinity-Instruct	4.45	3.38	3.91	5.01	3.93	4.50	5.45	5.16	5.45	4.92	4.41
	BELLE-SFT (BELLEGroup, 2023)	4.29	3.34	3.82	4.73	4.28	4.76	5.37	5.47	5.35	4.99	4.41
	TaP-SFT (GPT-4)	5.41	3.74	4.57	**5.24**	**4.93**	**5.19**	5.67	**5.89**	**5.82**	**5.46**	5.02
	TaP-SFT (DeepSeek-V2)	**5.58**	**4.20**	**4.89**	5.15	4.86	4.90	**5.75**	5.84	**5.82**	5.39	**5.14**
Qwen2.5-7B	Firefly	3.91	3.34	3.62	4.67	4.41	5.00	4.92	3.95	4.89	4.64	4.13
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.07	3.99	4.53	5.35	4.60	5.31	5.11	5.11	5.35	5.14	4.83
	COIG (Zhang et al., 2023a)	3.35	2.87	3.11	4.57	4.24	4.13	3.87	4.79	4.70	4.38	3.75
	MOSS-SFT (Sun et al., 2024)	5.09	4.01	4.55	5.30	4.47	4.85	5.16	5.76	5.50	5.17	4.86
	COIG-CQIA (Bai et al., 2024)	5.12	3.90	4.51	4.98	5.09	4.81	4.96	5.26	5.01	5.02	4.76
	Infinity-Instruct	4.99	4.41	4.70	5.22	5.00	5.71	5.67	5.79	5.87	5.54	5.12
	BELLE-SFT (BELLEGroup, 2023)	4.73	3.59	4.16	5.10	4.74	5.59	5.37	5.68	5.64	5.35	4.76
	TaP-SFT (GPT-4)	**6.84**	**5.36**	**6.10**	**6.06**	5.14	5.58	6.04	6.05	6.09	5.83	**5.96**
	TaP-SFT (DeepSeek-V2)	6.63	4.91	5.77	5.97	**5.53**	**5.76**	**6.07**	**6.11**	**6.13**	**5.93**	5.85
Llama-3.1-8B	Firefly	2.30	2.50	2.40	3.90	3.76	4.09	4.71	3.79	4.21	4.07	3.24
	Alpaca-GPT-4-ZH (Peng et al., 2023)	2.35	2.93	2.64	3.95	3.50	3.88	4.49	4.34	4.56	4.12	3.38
	COIG (Zhang et al., 2023a)	1.68	2.08	1.88	3.01	2.38	2.78	2.84	3.71	3.52	3.04	2.46
	MOSS-SFT (Sun et al., 2024)	2.51	2.87	2.69	4.10	3.09	4.04	4.57	5.37	4.74	4.32	3.50
	COIG-CQIA (Bai et al., 2024)	2.24	2.53	2.39	3.40	2.93	3.43	3.45	4.18	3.97	3.56	2.97
	Infinity-Instruct	2.88	3.14	3.01	4.71	**4.24**	4.38	**5.48**	5.26	**5.60**	**4.95**	3.98
	BELLE-SFT (BELLEGroup, 2023)	3.06	3.41	3.24	**4.78**	3.97	**4.62**	5.04	**5.55**	5.34	4.88	4.06
	TaP-SFT (GPT-4)	3.31	3.48	3.40	4.73	3.14	4.46	5.32	5.34	5.37	4.73	**4.06**
	TaP-SFT (DeepSeek-V2)	**3.44**	**3.50**	**3.47**	4.40	3.12	4.49	5.09	5.21	5.34	4.61	4.04
Gemma-2-9B	Firefly	2.29	2.59	2.44	3.81	3.64	4.24	4.77	3.53	4.40	4.06	3.25
	Alpaca-GPT-4-ZH (Peng et al., 2023)	3.18	3.21	3.19	4.24	2.76	4.25	4.47	4.74	4.59	4.17	3.68
	COIG (Zhang et al., 2023a)	1.64	2.26	1.95	2.90	1.95	2.46	2.56	3.42	3.53	2.80	2.38
	MOSS-SFT (Sun et al., 2024)	2.62	2.85	2.73	3.94	3.02	3.70	4.53	5.50	4.46	4.19	3.46
	COIG-CQIA (Bai et al., 2024)	2.49	2.79	2.64	3.26	2.74	3.75	3.45	4.00	4.14	3.56	3.10
	Infinity-Instruct	3.64	3.50	3.57	4.77	3.79	4.88	**5.33**	5.29	**5.85**	4.99	4.28
	BELLE-SFT (BELLEGroup, 2023)	3.80	**3.57**	**3.68**	**4.96**	**4.12**	**4.99**	5.25	**5.53**	5.55	**5.07**	**4.37**
	TaP-SFT (GPT-4)	**4.00**	3.20	3.60	4.84	3.33	4.76	5.25	5.26	5.81	4.88	4.24
	TaP-SFT (DeepSeek-V2)	3.84	3.38	3.61	4.51	3.16	4.60	5.08	5.03	5.13	4.58	4.10
Qwen2.5-14B	Firefly	4.40	3.79	4.10	4.81	5.10	5.06	5.13	4.26	4.97	4.89	4.49
	Alpaca-GPT-4-ZH (Peng et al., 2023)	5.89	4.79	5.34	5.77	5.41	5.88	5.36	6.05	5.90	5.73	5.54
	COIG (Zhang et al., 2023a)	3.50	3.25	3.38	4.91	4.62	4.95	4.41	5.61	4.68	4.86	4.12
	MOSS-SFT (Sun et al., 2024)	5.03	4.89	4.96	5.47	4.79	5.15	5.71	5.89	5.89	5.48	5.22
	COIG-CQIA (Bai et al., 2024)	5.40	4.47	4.94	5.48	5.00	5.55	5.17	5.13	5.21	5.26	5.10
	Infinity-Instruct	5.71	4.61	5.16	5.81	5.09	6.09	6.01	5.71	6.12	5.81	5.48
	BELLE-SFT (BELLEGroup, 2023)	4.72	4.37	4.55	5.56	4.74	6.13	5.55	5.89	5.78	5.61	5.08
	TaP-SFT (GPT-4)	6.79	**5.75**	6.27	**6.46**	5.78	6.49	**6.27**	**6.29**	**6.37**	**6.27**	6.27
	TaP-SFT (DeepSeek-V2)	**7.36**	5.57	**6.46**	6.37	**5.81**	**6.63**	6.17	6.24	6.22	6.24	**6.35**

Table 11: Performance comparison of five LLMs on Alignbench after supervised fine-tuning with various datasets. The evaluation is conducted using **DeepSeek-V3**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.

Model	Dataset	Reasoning			Language							Overall
		Math.	**Logi.**	**Avg.**	**Pro.**	**Chi.**	**Fund.**	**Writ.**	**Open.**	**Role.**	**Avg.**	
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.63	3.42	4.02	4.86	4.74	4.87	5.55	5.61	5.74	5.23	4.63
	Chinese-DPO-Pairs	5.09	3.84	4.46	5.08	**5.26**	4.97	6.01	6.00	5.78	5.52	4.99
	TaP (GPT-4)	**5.78**	**4.60**	**5.19**	**5.71**	5.25	**5.27**	**6.51**	**6.29**	**6.41**	**5.90**	**5.55**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.01	4.29	5.15	5.64	5.49	5.37	6.23	6.21	5.98	5.82	5.49
	Chinese-DPO-Pairs	6.09	4.41	5.25	5.74	**5.62**	**5.47**	6.45	**6.55**	6.20	6.01	5.63
	TaP (GPT-4)	**6.39**	**4.74**	**5.57**	**5.77**	5.40	5.46	**6.68**	6.53	**6.54**	**6.06**	**5.81**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	5.04	4.01	4.53	5.48	5.02	5.24	5.43	5.68	5.50	5.39	4.96
	Chinese-DPO-Pairs	5.36	4.40	4.88	5.54	5.18	5.36	5.75	**5.89**	5.49	5.53	5.21
	TaP (GPT-4)	**5.64**	**4.59**	**5.11**	**5.63**	**5.28**	**5.46**	**6.23**	5.79	**6.05**	**5.74**	**5.43**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	7.44	5.04	6.24	6.31	5.78	6.34	6.41	6.79	6.39	6.34	6.29
	Chinese-DPO-Pairs	7.46	**5.53**	6.49	6.52	5.91	6.15	6.64	6.79	6.53	6.42	6.46
	TaP (GPT-4)	**7.66**	5.52	**6.59**	**6.79**	**6.00**	**6.40**	**6.88**	**6.95**	**6.93**	**6.66**	**6.62**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.68	3.37	3.52	4.98	4.43	4.78	5.40	5.74	5.83	5.19	4.36
	Chinese-DPO-Pairs	3.81	3.25	3.53	5.06	4.45	**5.12**	6.15	**6.58**	6.04	5.57	4.55
	TaP (GPT-4)	**4.25**	**3.67**	**3.96**	**5.13**	**4.72**	4.94	**6.40**	6.03	**6.48**	**5.62**	**4.79**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.47	3.93	3.70	4.84	3.93	4.87	5.71	6.18	5.78	5.22	4.46
	Chinese-DPO-Pairs	**4.19**	3.95	4.07	**5.05**	3.93	4.99	6.00	6.05	6.14	5.36	4.71
	TaP (GPT-4)	4.14	**4.07**	**4.10**	5.02	**3.95**	**5.25**	**6.23**	**6.34**	**6.23**	**5.50**	**4.80**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.36	3.51	3.93	5.01	4.10	5.15	5.57	5.55	5.93	5.22	4.58
	Chinese-DPO-Pairs	4.18	3.54	3.86	5.17	4.29	5.10	5.67	5.74	5.97	5.32	4.59
	TaP (GPT-4)	**4.80**	**4.29**	**4.55**	**5.52**	**4.69**	**5.72**	**6.28**	**6.50**	**6.67**	**5.90**	**5.22**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	4.78	3.59	4.18	4.94	3.83	5.09	5.71	5.95	5.84	5.23	4.70
	Chinese-DPO-Pairs	5.12	**4.22**	4.67	5.23	**4.00**	**5.46**	6.11	**6.11**	6.23	5.52	5.09
	TaP (GPT-4)	**5.31**	4.12	**4.72**	**5.35**	3.98	**5.46**	**6.43**	**6.11**	**6.34**	**5.61**	**5.16**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.17	5.34	5.75	6.19	5.74	6.18	5.61	5.76	5.90	5.90	5.82
	Chinese-DPO-Pairs	6.21	5.49	5.85	6.27	5.78	6.03	5.80	6.05	5.84	5.96	5.90
	TaP (GPT-4)	**7.11**	**5.78**	**6.44**	**6.67**	**6.19**	**6.38**	**6.24**	**6.39**	**6.44**	**6.39**	**6.42**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	7.54	6.35	6.94	6.95	6.38	6.75	6.55	6.68	6.51	6.64	6.79
	Chinese-DPO-Pairs	7.93	6.05	6.99	**7.14**	6.31	**6.85**	6.93	7.05	6.84	6.85	6.92
	TaP (GPT-4)	**8.03**	**6.51**	**7.27**	7.09	**6.83**	6.59	**7.13**	**7.55**	**7.29**	**7.08**	**7.17**

Table 12: Performance comparison of LLMs trained with DPO using different datasets on AlignBench. The model names include two possible suffixes: "Open" and "TaP." The "Open" suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas "TaP" denotes initialization from models trained on a dataset constructed by TaP. The evaluation is conducted using **GPT-4o**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.

Model	Dataset	Reasoning			Language							Overall
		Math.	Logi.	Avg.	Pro.	Chi.	Fund.	Writ.	Open.	Role.	Avg.	
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.45	3.30	3.88	4.75	4.21	4.72	5.48	5.45	5.64	5.04	4.46
	Chinese-DPO-Pairs	4.69	3.71	4.20	4.77	**4.83**	4.71	5.67	5.66	5.82	5.24	4.72
	TaP (GPT-4)	**5.10**	**4.49**	**4.79**	**5.56**	4.69	**5.15**	**6.28**	**5.87**	**6.20**	**5.62**	**5.21**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	5.45	3.85	4.65	**5.45**	**5.09**	**5.33**	5.87	5.55	5.74	5.50	5.08
	Chinese-DPO-Pairs	5.61	**4.20**	**4.90**	5.44	4.64	5.25	5.89	5.63	**6.18**	5.51	5.20
	TaP (GPT-4)	**5.79**	4.02	**4.90**	5.27	4.88	5.28	**6.40**	**5.95**	6.14	**5.65**	**5.28**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.86	3.88	4.37	**5.52**	4.66	5.19	5.43	5.84	5.40	5.34	4.85
	Chinese-DPO-Pairs	5.15	4.13	4.64	5.47	4.57	5.27	5.75	5.84	5.51	5.40	5.02
	TaP (GPT-4)	**5.42**	**4.32**	**4.87**	5.35	**5.09**	**5.28**	**5.91**	**5.92**	**5.92**	**5.58**	**5.22**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.79	4.88	5.83	5.86	5.36	**6.34**	6.09	6.11	6.26	6.00	5.92
	Chinese-DPO-Pairs	6.87	5.01	5.94	6.03	**5.52**	6.16	6.13	6.16	6.51	6.09	6.01
	TaP (GPT-4)	**6.92**	**5.14**	**6.03**	**6.28**	5.26	6.24	**6.43**	**6.32**	**6.59**	**6.18**	**6.11**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.29	3.29	3.29	4.71	3.97	4.65	5.52	5.63	5.88	5.06	4.17
	Chinese-DPO-Pairs	3.49	3.10	3.29	4.91	4.05	**4.87**	5.85	**6.00**	6.08	5.29	4.29
	TaP (GPT-4)	**3.58**	**3.41**	**3.50**	**4.95**	**4.29**	4.76	**6.28**	5.66	**6.28**	**5.37**	**4.43**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.01	**3.64**	3.33	4.61	3.22	4.65	5.39	5.61	5.71	4.86	4.09
	Chinese-DPO-Pairs	**3.54**	3.63	**3.59**	**4.73**	**3.69**	**4.71**	5.59	5.66	6.08	**5.07**	**4.33**
	TaP (GPT-4)	3.46	3.58	3.52	4.52	3.26	4.49	**5.73**	**5.82**	**6.10**	4.99	4.25
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.72	3.35	3.54	4.81	3.72	4.99	5.49	5.39	5.90	5.05	4.29
	Chinese-DPO-Pairs	3.77	3.46	3.61	5.09	3.83	5.09	5.60	5.76	5.87	5.21	4.41
	TaP (GPT-4)	**4.16**	**4.01**	**4.09**	**5.17**	**3.98**	**5.32**	**5.96**	**5.87**	**6.33**	**5.44**	**4.76**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	4.37	3.17	3.77	4.69	**3.43**	5.25	5.32	5.34	5.56	4.93	4.35
	Chinese-DPO-Pairs	**4.54**	**3.64**	**4.09**	**5.15**	**3.43**	**5.41**	5.57	5.42	**6.24**	**5.21**	**4.65**
	TaP (GPT-4)	4.49	3.61	4.05	4.85	3.41	5.18	**5.88**	**5.55**	5.99	5.14	4.60
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.12	5.17	5.64	5.98	5.64	5.90	5.51	5.87	5.97	5.81	5.73
	Chinese-DPO-Pairs	6.05	4.89	5.47	6.04	5.43	5.88	5.63	5.87	5.91	5.79	5.63
	TaP (GPT-4)	**6.77**	**5.34**	**6.05**	**6.37**	**5.74**	**6.26**	**6.04**	**6.21**	**6.41**	**6.17**	**6.11**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	7.19	5.64	6.41	6.41	5.83	6.60	6.19	6.24	6.43	6.28	6.35
	Chinese-DPO-Pairs	**7.44**	5.38	6.41	**6.57**	5.79	**6.65**	6.51	6.32	6.74	6.43	6.42
	TaP (GPT-4)	7.31	**5.73**	**6.52**	6.43	**5.91**	6.63	**6.77**	**6.42**	**6.96**	**6.52**	**6.52**

Table 13: Performance comparison of LLMs trained with DPO using different datasets on AlignBench. The model names include two possible suffixes: "Open" and "TaP." The "Open" suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas "TaP" denotes initialization from models trained on a dataset constructed by TaP. The evaluation is conducted using **DeepSeek-V3**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.

Model	Dataset	Reasoning			Language							Overall
		Math.	Logi.	Avg.	Pro.	Chi.	Fund.	Writ.	Open.	Role.	Avg.	
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.62	3.65	4.13	4.45	4.71	4.57	5.03	5.18	5.36	4.88	4.51
	Chinese-DPO-Pairs	5.82	**4.42**	5.12	**5.83**	5.22	5.07	6.43	**6.89**	6.41	5.98	5.55
	TaP (GPT-4)	**6.24**	4.16	**5.20**	5.73	**5.50**	**5.61**	**6.49**	6.71	**6.60**	**6.11**	**5.66**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	5.59	3.63	4.61	4.97	4.79	5.07	5.48	5.79	5.53	5.27	4.94
	Chinese-DPO-Pairs	5.82	3.93	4.88	5.17	5.41	5.13	5.68	5.97	5.71	5.51	5.20
	TaP (GPT-4)	**6.49**	**4.89**	**5.69**	**5.93**	**6.12**	**5.53**	**6.55**	**6.89**	**6.59**	**6.27**	**5.98**
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	5.04	4.39	4.71	5.60	5.00	**5.68**	5.61	5.92	5.53	5.56	5.13
	Chinese-DPO-Pairs	5.03	3.79	4.41	5.15	5.18	5.28	5.72	5.32	5.34	5.33	4.87
	TaP (GPT-4)	**5.29**	**4.60**	**4.94**	**5.66**	**5.31**	5.50	**6.28**	**6.29**	**5.83**	**5.81**	**5.38**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	**7.11**	5.38	6.24	6.45	**6.19**	6.26	6.40	6.89	6.46	6.44	6.34
	Chinese-DPO-Pairs	6.82	4.75	5.79	5.73	5.67	5.53	5.76	7.11	6.34	6.02	5.90
	TaP (GPT-4)	7.05	**5.53**	**6.29**	**6.77**	5.91	**6.35**	**6.81**	**7.13**	**6.75**	**6.62**	**6.46**
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.78	**3.66**	3.72	5.02	4.36	**4.91**	5.67	5.92	5.74	5.27	4.50
	Chinese-DPO-Pairs	2.76	2.83	2.79	4.26	3.48	4.46	4.83	4.82	5.29	4.52	3.66
	TaP (GPT-4)	**3.90**	3.65	**3.78**	**5.17**	**4.67**	4.88	**6.31**	**6.21**	**6.20**	**5.57**	**4.68**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.81	4.08	3.94	4.81	4.21	5.09	5.73	5.89	5.78	5.25	4.60
	Chinese-DPO-Pairs	3.56	3.88	3.72	5.00	4.05	4.88	5.73	5.50	5.35	5.09	4.40
	TaP (GPT-4)	**3.92**	**4.15**	**4.04**	**5.05**	**4.24**	**5.12**	**6.33**	**6.53**	**6.29**	**5.59**	**4.81**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.23	3.26	3.75	5.09	3.91	4.44	4.88	5.03	5.16	4.75	4.25
	Chinese-DPO-Pairs	4.38	3.73	4.05	4.89	4.63	4.97	6.15	6.24	6.01	5.48	4.77
	TaP (GPT-4)	**4.85**	**4.14**	**4.49**	**5.75**	**4.66**	**5.45**	**6.69**	**6.34**	**6.45**	**5.89**	**5.19**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	4.13	3.57	3.85	5.04	**4.53**	5.13	5.53	5.26	5.73	5.20	4.53
	Chinese-DPO-Pairs	4.52	3.77	4.14	5.02	4.12	**5.24**	5.48	6.05	5.98	5.32	4.73
	TaP (GPT-4)	**4.83**	**4.00**	**4.42**	**5.35**	4.14	5.15	**6.14**	**6.47**	**6.24**	**5.58**	**5.00**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.03	5.18	5.61	5.50	6.05	6.35	5.73	5.82	5.93	5.90	5.75
	Chinese-DPO-Pairs	6.60	5.51	6.05	6.58	5.81	**6.46**	6.27	6.42	6.24	6.30	6.18
	TaP (GPT-4)	**7.10**	**5.96**	**6.53**	**6.77**	**6.41**	6.06	**6.52**	**6.74**	**6.43**	**6.49**	**6.51**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.79	6.43	6.61	6.07	6.38	6.22	6.59	7.03	6.69	6.50	6.55
	Chinese-DPO-Pairs	7.32	5.42	6.37	6.22	6.22	6.59	6.43	6.58	6.34	6.40	6.38
	TaP (GPT-4)	**8.07**	**6.76**	**7.42**	**7.30**	**6.74**	**6.81**	**7.19**	**7.42**	**7.27**	**7.12**	**7.27**

Table 14: Performance comparison of LLMs trained with PPO using different datasets on AlignBench. The model names include two possible suffixes: "Open" and "TaP." The "Open" suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas "TaP" denotes initialization from models trained on a dataset constructed by TaP. The evaluation is conducted using **GPT-4o**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.

Model	Dataset	Reasoning			Language							Overall
		Math.	Logi.	Avg.	Pro.	Chi.	Fund.	Writ.	Open.	Role.	Avg.	
Qwen2.5-3B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.93	**3.96**	4.44	5.10	**4.81**	4.72	5.68	5.76	6.07	5.36	4.90
	Chinese-DPO-Pairs	5.14	3.91	4.53	**5.31**	4.64	4.85	6.12	**6.13**	6.06	5.52	5.02
	TaP (GPT-4)	**5.61**	3.82	**4.71**	5.29	4.71	**5.21**	**6.16**	5.97	**6.40**	**5.62**	**5.17**
Qwen2.5-3B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	5.33	3.90	4.62	5.28	4.95	**5.37**	5.81	5.89	6.00	5.55	5.08
	Chinese-DPO-Pairs	5.83	**4.30**	**5.07**	**5.57**	**5.43**	5.25	**6.27**	5.92	**6.16**	**5.77**	**5.42**
	TaP (GPT-4)	**5.89**	4.07	4.98	5.53	5.32	5.18	5.95	**6.03**	6.08	5.68	5.33
Qwen2.5-7B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	4.85	4.08	4.46	5.19	4.55	**5.60**	5.67	**5.84**	5.72	5.43	4.95
	Chinese-DPO-Pairs	4.82	3.79	4.31	**5.48**	**4.93**	5.45	**5.69**	5.79	**5.78**	**5.52**	4.91
	TaP (GPT-4)	**5.03**	**4.25**	**4.64**	5.39	4.84	5.09	5.68	5.82	5.66	5.41	**5.03**
Qwen2.5-7B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.67	4.67	5.67	6.04	5.52	5.69	6.03	6.26	6.32	5.98	5.82
	Chinese-DPO-Pairs	**6.87**	4.92	**5.89**	5.97	**5.62**	5.85	6.20	6.29	6.34	6.04	**5.97**
	TaP (GPT-4)	6.51	**4.95**	5.73	**6.36**	5.26	**6.21**	**6.25**	**6.35**	**6.49**	**6.15**	5.94
Llama-3.1-8B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	**3.54**	**3.27**	**3.41**	4.81	3.79	4.43	5.16	5.82	5.80	4.97	4.19
	Chinese-DPO-Pairs	3.10	3.11	3.10	4.91	3.83	**4.78**	5.49	5.26	5.88	5.03	4.06
	TaP (GPT-4)	3.43	3.23	3.33	**4.99**	**4.10**	4.40	**6.09**	**5.97**	**6.12**	**5.28**	**4.30**
Llama-3.1-8B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	**3.50**	**3.73**	**3.61**	4.28	**3.66**	4.53	5.36	5.50	5.81	4.86	4.24
	Chinese-DPO-Pairs	3.13	3.47	3.30	4.47	3.48	**4.69**	5.55	5.55	5.59	4.89	4.09
	TaP (GPT-4)	3.17	3.68	3.43	**4.52**	3.31	4.63	**5.60**	**5.66**	**5.90**	**4.94**	**4.18**
Gemma-2-9B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	3.98	3.63	3.81	4.98	4.00	4.99	5.25	5.92	5.71	5.14	4.47
	Chinese-DPO-Pairs	4.06	3.39	3.73	4.84	**4.14**	4.87	5.88	5.68	6.09	5.25	4.49
	TaP (GPT-4)	**4.26**	**3.65**	**3.96**	**5.59**	3.84	**5.37**	**6.16**	**6.08**	**6.24**	**5.55**	**4.75**
Gemma-2-9B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	3.62	3.50	3.56	4.69	**3.83**	**4.96**	5.17	5.50	5.61	4.96	4.26
	Chinese-DPO-Pairs	3.82	3.28	3.55	4.69	3.28	4.93	5.40	5.42	5.73	4.91	4.23
	TaP (GPT-4)	**4.31**	**3.51**	**3.91**	**4.98**	3.67	**4.96**	**5.80**	**5.92**	**6.07**	**5.23**	**4.57**
Qwen2.5-14B-SFT-Open	Huozi-RLHF (Huozi-Team, 2024)	6.53	4.86	5.69	5.83	5.71	6.25	5.64	6.03	6.10	5.93	5.81
	Chinese-DPO-Pairs	6.35	**5.35**	5.85	**6.22**	5.62	**6.50**	6.08	**6.32**	6.29	**6.17**	6.01
	TaP (GPT-4)	**6.73**	5.32	**6.02**	6.13	**5.79**	6.00	**6.28**	**6.32**	**6.43**	6.16	**6.09**
Qwen2.5-14B-SFT-TaP	Huozi-RLHF (Huozi-Team, 2024)	6.94	5.90	6.42	6.55	5.88	6.58	6.17	6.34	6.49	6.34	6.38
	Chinese-DPO-Pairs	7.23	5.73	6.48	**6.68**	**6.14**	**6.91**	6.57	**6.50**	6.72	**6.59**	6.53
	TaP (GPT-4)	**7.38**	**6.29**	**6.84**	6.67	6.05	6.35	**6.64**	6.45	**6.81**	6.50	**6.67**

Table 15: Performance comparison of LLMs trained with PPO using different datasets on AlignBench. The model names include two possible suffixes: "Open" and "TaP." The "Open" suffix indicates that the LLMs were initialized from models trained via supervised fine-tuning on open-source datasets, whereas "TaP" denotes initialization from models trained on a dataset constructed by TaP. The evaluation is conducted using **DeepSeek-V3**, which scores the models' responses. The "Fund." column denotes Fundamental Language Ability, "Chi." denotes Advanced Chinese Understanding, "Open." denotes Open-ended Questions, "Writ." denotes Writing Ability, "Role." denotes Task-oriented Role Play, "Pro" denotes "Professional Knowledge", "Math." denotes Mathematics, and "Logic." denotes Logical Reasoning.