

# 🐼 PANDA: Preference Adaptation for Enhancing Domain-Specific Abilities of LLMs

Anonymous ACL submission

## Abstract

While Large language models (LLMs) have demonstrated considerable capabilities across various natural language tasks, they often fall short of the performance achieved by domain-specific state-of-the-art models. One potential approach to enhance domain-specific capabilities of LLMs involves fine-tuning them using corresponding datasets. However, this method can be both resource and time-intensive, and not applicable to closed-source commercial LLMs. In this paper, we propose **Preference Adaptation for Enhancing Domain-specific Abilities of LLMs (PANDA)**, a method designed to augment the domain-specific capabilities of LLMs by leveraging insights from the response preference of expert models without requiring fine-tuning. Our experimental results reveal that PANDA significantly enhances the domain-specific ability of LLMs on text classification and interactive decision tasks. Moreover, LLM with PANDA even outperforms the expert model that being learned on 4 tasks of ScienceWorld. This finding highlights the potential of exploring tuning-free approaches to achieve weak-to-strong generalization.

## 1 Introduction

Large language models (LLMs) have shown exceptional performance across a broad spectrum of tasks via prompting (Brown et al., 2020; Ouyang et al., 2021; OpenAI, 2023; Touvron et al., 2023; Jiang et al., 2024), suggesting they possess advanced general-purpose capabilities. Despite this, when tested on specific tasks, the effectiveness of these general-purpose LLMs often falls short of specialized state-of-the-art models (Kocón et al., 2023; Zhong et al., 2023; Lin et al., 2023). Therefore, improving the domain-specific abilities of general-purpose LLMs remains a critical challenge.

To address this challenge, one straightforward method is Knowledge Distillation (Hinton et al., 2015, KD), where the LLM acts as the student

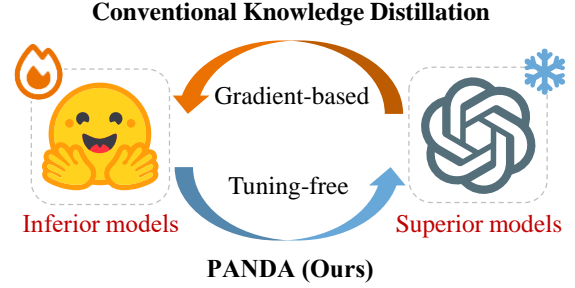


Figure 1: PANDA aims to enhance domain-specific capability of LLMs, which possess superior general capability, by learning from domain expert models that have inferior general capability with a tuning-free way. While conventional knowledge distillation usually leverage superior models to teach inferior models via gradient-based methods. The direction of the arrow represents the direction of knowledge transfer.

model and the domain expert model acts as the teacher model. However, it can be both resource-intensive and time-consuming. It might also inevitably reduce the original capabilities of the LLM, and in severe cases, could significantly impair them, as noted by Qi et al. (2023). Moreover, some of the most sophisticated LLMs (OpenAI, 2023; GeminiTeam et al., 2023) either do not offer a fine-tuning interface or provide only a limited set of fine-tuning options, rendering KD impractical. Therefore, KD is not an effective solution for this challenge.

Fortunately, tuning-free methods shed light on resolving the problem. Two notable classes of tuning-free methods are Retrieval-Augmentation-Generation (RAG) and self-reflection. RAG retrieves context information from a database during inference to enhance the prompt, thereby improving the performance of LLMs (Gao et al., 2023). However, previous RAG methods are not specifically designed to learn from expert models and do not fully utilize the capabilities of these models. On the other hand, self-reflection-based methods utilize LLMs to refine their outputs based on feedback

from the environment (Shinn et al., 2023; Wang and Li, 2023), ground-truth via self-reflection (Yang et al., 2023), or feedback from domain expert models (Lin et al., 2023; Lu et al., 2023). Despite their cost-efficiency due to being tuning-free, these methods often fall short of achieving the same level of performance as state-of-the-art (SOTA) models in specific domains (Lin et al., 2023; Zhong et al., 2023; Lai et al., 2023). Additionally, the need to deploy an extra expert model during inference and the LLMs’ limited ability to understand expert knowledge impose limitations on the potential performance improvement. Therefore, effectively leveraging expert models to fully equip LLMs with the capacity for specific tasks remains a critical issue to address.

To this end, we propose the **Preference Adaptation for Enhancing Domain-specific Abilities of LLMs (PANDA)** method, a tuning-free approach for enhancing the domain-specific capabilities of LLMs. To achieve this, we initially task a domain expert model to infer on the training data and extract their output samples, which serve as expert preferences. Subsequently, we prompt the LLM to generate explanations (referred to as “insights”) for the preferences of the expert model and gather these queries and insights into an “insight pool”. This approach allows the LLM to develop a deeper understanding of the implicit knowledge provided by the expert model beyond mere surface behavior. During inference, PANDA retrieves the most relevant insights from the insight pool to assist in completing the current query, followed by standard inference using the retrieved insights. Compared to RAG, our work focuses on constructing a comprehensive knowledge base to better support LLMs in completing specific tasks. In contrast to self-reflection, our work emphasizes leveraging expert knowledge more effectively, rather than directly relying on the expert model, with the goal of achieving a higher ceiling performance.

We conduct comprehensive experiments on interactive decision making and text classification tasks to evaluate the effectiveness of PANDA. Our experimental results demonstrate that PANDA enhances the domain-specific capability of LLMs across 16 tasks in ScienceWorld (Wang et al., 2022) and 7 tasks in TweetEval (Barbieri et al., 2020). In summary, our contributions are as follows:

- To the best of our knowledge, we are the first

to explore how to assist LLMs in effectively learning from expert models.

- We propose PANDA as a tuning-free approach that enhances the domain-specific capabilities of LLMs by adapting their preferences to align with the expert.
- Extensive experimental results demonstrate that PANDA significantly improves the domain-specific capabilities of LLMs across 16 tasks in ScienceWorld and 7 tasks in TweetEval. Particularly, LLMs with PANDA even outperforms the expert model that being learned on 4 tasks in ScienceWorld, revealing the potential of tuning-free approaches to achieve weak-to-strong generalization (Burns et al., 2023).

## 2 Preference Adaptation for Enhancing Domain-Specific Abilities of LLMs

### 2.1 Overview

As shown in Fig. 2, PANDA consists of two main stages: the learning stage and the inference stage. The learning stage aims to capture the expert knowledge comprehensively. During this stage, the expert model is utilized to generate samples on the training dataset. PANDA then constructs preference pairs for each data instance based on specific preference heuristics. The preference heuristics may vary depending on the type of expert model being used. For a classification model, the preference rank can be determined based on the output logits distribution of each class, where higher logits may indicate that the expert model favors considering the specific query as belonging to that class..

Next, the LLM is prompted with the preference information to generate explanations. These explanations, derived from the training dataset, form the insight pool. During inference, the insight pool is utilized to retrieve relevant insights that can guide the LLM in adapting its preferences to align with those of the expert model. The goal is to enhance the domain-specific capability of the LLM by aligning its preferences with those of the expert model.

Meanwhile, it is worth noting that although PANDA and conventional KD differ significantly in terms of implementation techniques, they share a similar essence in each stage. We show discussion details in Section 2.4.

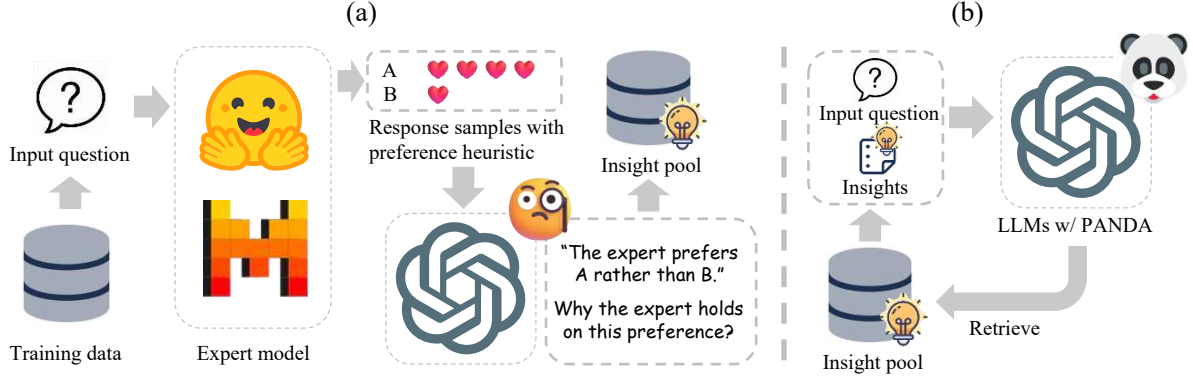


Figure 2: PANDA consists of two main stages: (a) the learning stage acquires insights from expert preferences and forms an insight pool; (b) the inference stage retrieves relevant insights from the insight pool and perform preference adaptation via in-context learning.

## 2.2 Learning from Expert Preferences

When it comes to how to let LLMs learn from the domain expert models, a naive approach involves merely memorizing the behavior demonstrated by the expert models. A better approach involves not only memorizing the desired behavior but also considering the undesirable behaviors that the expert avoids in order to avoid making mistakes. Moreover, the optimal approach goes beyond mere memorization of behavior; it also seeks to understand the underlying rationale behind actions of the expert. Drawing inspiration from this optimal learning process, we propose to learn from expert to boost the domain-specific performance of LLMs.

Inspired by the success of RLHF (Ziegler et al., 2019), which firstly trains a reward model using human preference data and then leverage it to perform reinforcement learning for LLMs to reach good alignment with humans. As Fig. 2-(a) shows, we propose to learning from expert preferences as well since preferences information represents more comprehensive expert knowledge compared to solely behavior information.

Leveraging the powerful language understanding and generation ability, we prompt LLMs to generate the explanation of the preferences of the expert, which is a process of learning expert knowledge. The simplified prompt template is as follows:

### Prompt for PANDA-Learning

```
{Query}
The expert prefer {A} rather than {B}. Explain the reason why the expert holds on this preferences.
```

Note that A and B here represent a pair of responses from the expert and its confidence in A is greater than it in B, which indicates the expert has a stronger preference for responding with A rather than B when presented with the Query. We refer to the response of LLMs to the prompt for learning as Expert Insight, which is then utilized to adapt the preferences of the LLMs towards better performance in completing the specific task.

## 2.3 Preference Adaptation during Inference

Motivated by the intuition that strategies employed for similar problems often result in mutual benefits, we consider the problems with similar context as similar problems and insights from similar problems are considered relevant. As illustrated in Fig. 2-(b), when encountering a new query, PANDA firstly retrieves related insights from the insight pool using the current query context as the retrieval key. In contrast to self-reflection-based methods, LLMs equipped with PANDA can achieve improved performance without the need for trial-and-error, thereby reducing the potential interaction cost.

Benefiting from the in-context learning ability of LLMs, we then incorporate the retrieved insights as part of the prompt, as described below:

### Prompt for PANDA-Inference

```
{prompt Context}
These are some insights that may be helpful for you to improve success rate:
{retrieved insights}
{prompt Context}
```

Compared to vanilla RAG-based methods that retrieve in-context examples, PANDA leverages insights from the expert preferences generated by LLMs, resulting in improved generalization.

## 2.4 Connection between PANDA and Conventional Knowledge Distillation

PANDA and conventional knowledge distillation (KD) differ significantly in terms of implementation details. PANDA is a tuning-free approach, whereas conventional KD typically requires performing gradient descent to enable the student model to mimic the capabilities of the teacher model. Additionally, in conventional KD, the teacher model is usually superior to the student model, which is the opposite of PANDA’s problem setting. PANDA aims to enhance domain-specific capability of LLMs by leveraging the expertise of the expert model, while benefiting from the strong foundational capabilities of LLMs.

Despite these differences in technical details, we can demonstrate that PANDA and conventional KD share conceptual similarities at each stage. During the learning stage, conventional KD employs gradient descent to minimize the KL divergence between the output distributions of the student and teacher models:

$$\theta = \arg \min_{\theta'} \sum_D \text{KL}(p_{\theta'}, p_t),$$

where  $\theta$  and  $\theta'$  are the weights of the student model learned from the teacher model.  $\text{KL}(\cdot, \cdot)$  is the KL divergence.  $p_{\theta'}$  and  $p_t$  represent the output distribution of the student model and teacher model respectively. This minimization aims to reduce the knowledge gap between the student and teacher models, implicitly enhancing the capability of the student model. In contrast, PANDA achieves a similar goal but in a more direct manner. To comprehensively capture the expert knowledge, PANDA leverages insights derived from the expert, rather than relying solely on the raw responses, to gain a deep understanding of the expert’s knowledge. By incorporating insights from expert preferences, PANDA aims to enhance the knowledge comprehensiveness, going beyond capturing the expert’s behavioral patterns alone.

PANDA also shares a similar essence with conventional KD during the inference stage. In conventional KD, the student model, with updated weights, is utilized for inference to achieve good

performance. Similarly, leveraging the in-context learning capability of LLMs, PANDA incorporates relevant insights obtained from the expert preferences into the prompt context. This integration aims to enhance the capability of LLMs to effectively complete the given task.

## 3 Experiments

To validate effectiveness of PANDA, We evaluate it on interactive decision making and text classification tasks. In all experiments, we utilize SentenceBERT (Reimers and Gurevych, 2019) to calculate sentence embeddings and retrieve relevant insights base on cosine similarity. We set the number of retrieved insights to 1 in ScienceWorld and 6 in TweetEval. All of the experiments take gpt-3.5-turbo-1106 as the LLM in PANDA.

### 3.1 Interactive Decision Making

**Benchmark.** In order to evaluate the effectiveness of PANDA in interactive decision-making tasks, we selected ScienceWorld (Wang et al., 2022) as the benchmark, which is an interactive text simulation of a laboratory environment, consisting of 10 sub-rooms such as Green House, Foundry, and Workshop. It provides a rich set of actions, with approximately 200k possible actions per step, including actions like moving to the greenhouse or picking up a jug, thereby posing challenges for LLMs in terms of their capabilities in reasoning, planning and grounding to the embodied environment. The ScienceWorld benchmark encompasses 30 tasks across 10 distinct classes. Each task exhibits significant variations, including differences in environment configurations (e.g., objects present in different rooms), specific task goals, and scientific domains (e.g., growing a plant, measuring the boiling point of substances). The number of steps required to complete each task varies, ranging from under 10 steps to over 100 steps. To ensure fairness and manage cost constraints, we randomly sampled 2 tasks from each task class, resulting in a total of 20 tasks for our evaluation.

**Baselines.** We compare PANDA with three baselines: ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023) and SayCan (Ahn et al., 2022), which have been proposed for grounding LLMs in agent tasks and enhancing their performance in complex interactive tasks. ReAct combines reasoning and acting in LLMs to enhance their ability to solve complex interactive tasks. Reflexion prompts



Task	Expert	ReAct	ReAct <sub>p</sub>	Reflexion	Reflexion <sub>p</sub>	SayCan	SayCan <sub>p</sub>
Task10-2	17.0	0.2	<u>2.1</u>	4.3	<u>10.2</u>	25.3	<u>32.4</u>
Task10-1	16.9	21.7	<u>38.9</u>	44.6	<u>39.4</u>	29.9	<u>33.5</u>
Task9-3	66.5	10.0	<u>12.8</u>	14.4	<u>18.9</u>	13.8	<u>18.0</u>
Task9-2	41.5	16.5	<u>21.7</u>	31.7	<u>38.5</u>	21.6	<u>24.1</u>
Task8-2	36.6	0.0	<u>2.3</u>	2.4	<u>6.2</u>	9.1	9.2
Task8-1	8.0	4.0	<u>7.0</u>	6.2	<u>6.4</u>	12.6	12.8
Task7-3	69.9	46.4	<b>81.0</b>	61.4	<b>75.2</b>	50.6	48.8
Task7-1	85.0	50.0	<b>95.5</b>	82.5	<u>72.0</u>	62.0	54.0
Task6-3	13.7	7.2	<u>10.4</u>	10.4	<u>11.5</u>	7.8	<u>8.6</u>
Task6-1	22.9	18.0	<b>26.4</b>	29.3	27.3	17.4	<u>18.1</u>
Task5-2	17.0	36.2	<u>23.7</u>	52.5	49.4	42.8	<u>50.1</u>
Task5-1	28.5	3.7	<u>5.5</u>	6.0	6.3	8.3	8.4
Task4-2	96.7	64.1	<u>72.5</u>	81.4	<u>85.0</u>	79.8	<u>80.7</u>
Task4-1	100.0	15.1	<u>26.6</u>	16.4	<u>28.4</u>	10.7	<u>14.3</u>
Task3-4	72.0	64.8	<b>76.3</b>	85.1	<u>75.5</u>	40.7	<u>46.7</u>
Task3-1	73.8	15.4	<u>22.8</u>	27.4	19.6	32.0	25.0
Task2-3	5.8	33.3	<u>37.7</u>	73.8	<u>77.6</u>	29.5	26.3
Task2-1	8.7	11.3	9.6	22.2	9.2	27.9	27.2
Task1-4	30.7	0.1	0.0	0.0	0.0	2.0	1.8
Task1-1	44.8	0.0	0.2	0.0	0.1	0.7	1.0
#n improvement	/	/	16	/	9	/	10

Table 1: Comparative results on ScienceWorld. The expert model is a fine-tuned flan-t5-large. ReAct<sub>p</sub>, Reflexion<sub>p</sub> and SayCan<sub>p</sub> represent the methods employed with PANDA. The term “#n improvement” refers to the count of tasks achieving improved result when implementing PANDA. PANDA boosts ReAct to achieve “weak-to-strong generalization” on 4 tasks (7-3, 7-1, 6-1, 3-4). We mark scores that achieve “weak-to-strong generalization” in **bold**, and scores that show improvement with PANDA in underline.

LLMs to generate reflections on their failure trajectory, providing verbal reinforcement feedback to aid in their self-improvement. SayCan leverages a value function that implicitly serves as a policy, utilizing grounding information about the environment to guide LLMs in task completion. We adopted the same implementation details as (Lin et al., 2023), with the exception that all our experiments utilize the gpt-3.5-turbo-1106 model. Furthermore, while maintaining rationality, we extended the original observation from the environment by concatenating it with information about the current rooms and inventory.

As the expert model, we utilize a fine-tuned flan-t5-large (Lin et al., 2023), which is the state-of-the-art single-model on ScienceWorld. It was fine-tuned using a multi-hop behavior cloning strategy, and the original dataset was downsampled to achieve good multi-task performance.

Prompt of PANDA-Inference for ScienceWorld

```
{Init Prompt}
These are some insights that may be helpful
for you to improve the success rate:
{Retrieved Insights}
{Current Trajectory}
```

For PANDA, we performed beam search and selected the top-2 responses from the expert model to construct the preference pair and learning prompt. The observation of the environment served as the retrieval key for the insight pool, inspired by the intuition that strategies employed for similar problems often yield mutual benefits.

**Results.** Since PANDA is agnostic to the underlying agent, we evaluate PANDA in comparison to each baseline. As shown in Table 1, even powerful agents like ReAct and SayCan fail to surpass the expert model. PANDA improves upon the ReAct baseline in 16 out of 20 tasks, the SayCan baseline in 10 tasks and even improves Reflexion in 9 tasks out of 20 tasks, which is an extremely strong baseline agent. Notably, ReAct with PANDA achieves a performance that surpasses the expert model in 4 tasks, demonstrating a similar phenomenon of *weak-to-strong generalization* observed in (Burns et al., 2023). Moreover, PANDA achieves these results without requiring any tuning, making it a promising approach towards achieving superalignment with LLMs. In Task 5-2 and Task 2-1, where ReAct outperforms the expert model, PANDA, in its effort to align the preferences of ReAct with the expert model, leads to a degradation in the performance of ReAct. It suggests that PANDA ad-

	Emoji	Emotion	Offensive	Abortion	Atheism	Climate	Sentiment
Expert	29.7	80.7	81.6	52.2	72.0	54.3	71.6
Zero-Shot w/ PANDA	8.4 <u>10.6</u>	65.7 61.1	69.6 68.9	55.6 <u>60.1</u>	35.4 <u>48.3</u>	63.6 <u>65.2</u>	63.0 <u>65.8</u>
Few-Shot w/ PANDA	8.8 <u>10.3</u>	74.1 <u>77.9</u>	74.2 72.9	58.4 <u>61.1</u>	25.6 <u>55.4</u>	50.3 <u>69.1</u>	65.9 <u>67.1</u>
ZS-CoT w/ PANDA	15.1 <u>20.1</u>	53.4 <u>61.1</u>	44.9 <u>47.1</u>	39.9 <u>47.7</u>	23.7 <u>42.5</u>	48.9 47.5	48.6 <u>50.1</u>
FS-CoT w/ PANDA	18.3 <u>19.0</u>	52.6 <u>57.5</u>	48.1 <u>48.8</u>	61.7 <u>66.3</u>	22.4 <u>28.2</u>	58.3 <u>63.2</u>	48.4 48.5

Table 2: Main results on 7 tasks from TweetEval. The expert model is fine-tuned RoBERTa-base models. The introduction of PANDA significantly enhances the performance of GPT-3.5-turbo across almost all tasks in four different settings: zero-shot, few-shot, Chain-of-Thought, and few-shot CoT. We mark scores that show improvement with PANDA in underline.

justs the preferences of ReAct to better conform to the preferences of the expert model, even in cases where the expert model may not perform well.

### 3.2 Text Classification

**Benchmark.** TweetEval (Barbieri et al., 2020) is a benchmark that consists of 11 heterogeneous Twitter-specific classification tasks, including Sentiment, Emoji, Emotion, Hate, Irony, Offensive and Stance. These tasks encompass both pragmatic and semantic aspects of tweet classification. It is worth noting that the distribution of samples across different classes in TweetEval tasks can be imbalanced. In our evaluation, we utilized the macro-averaged F1 score as the performance metric. Since PANDA aims to enhance the domain-specific capabilities of LLMs through learning from domain experts, in this section we primarily focus on presenting the results of tasks where *the expert model consistently outperforms LLMs*. This allows us to showcase the potential of PANDA in addressing the performance gap between LLMs and domain-specific fine-tuned models in these challenging tasks.

**Baselines.** As LLMs have strong instruction-following ability, we take zero-shot as a baseline and few-shot with 3 exemplars serving as a stronger baseline. We additionally test PANDA upon Chain-of-Thought (Wei et al., 2022) within zero-shot and few-shot settings, which output intermediate rationales to boost reasoning capability of LLMs.

We utilize the fine-tuned RoBERTa-base models<sup>1</sup>, which has been tuned on the training data for each task, as our domain expert model for GPT-3.5-turbo to learn from. We take the top-2 classes as

<sup>1</sup><https://huggingface.co/cardiffnlp/roberta-base-emotion>

the preference pairs data to learn from. To facilitate this learning process, we extract preference pairs data by sampling the top two classes for each task.

#### Prompt of PANDA-Inference for TweetEval

These are some insights that may be helpful for you to improve the success rate:  
{Retrieved Insights}  
{Zero-shot/ Few-shot/ CoT Prompt}

**Results.** As indicated in Table 2, GPT-3.5-turbo falls short of achieving the same level of performance as the fine-tuned roberta-base models on 7 tasks within the TweetEval dataset. Even when employing powerful prompt techniques like few-shot or Chain-of-Thought, the performance of the fine-tuned RoBERTa-base models still surpasses that of GPT-3.5-turbo on 5 tasks. However, the introduction of PANDA significantly enhances the performance of GPT-3.5-turbo across almost all tasks in four different settings: zero-shot, few-shot, Chain-of-Thought (CoT), and few-shot CoT. It is noteworthy that PANDA improves the performance of GPT-3.5-turbo on the climate and abortion stance classification tasks, where the expert model initially performed worse than GPT-3.5-turbo. This implies that PANDA not only incorporates the strengths of other models but also extracts valuable knowledge from their weaknesses, which can aid in self-improvement.

## 4 Ablation and Analysis

**PANDA elicits generalizable knowledge from preferences data.** We conducted an ablation study on 10 tasks of ScienceWorld benchmark to

Task	Expert	ReAct	PANDA 3	PANDA 2	PANDA 1	Raw 1	Raw 2
Task10-2	17.0	0.2	4.1	2.1	<b>6.4</b>	0.1	2.2
Task9-3	66.5	10.0	<b>29.4</b>	12.8	12.1	10.2	10.0
Task8-2	36.6	0.0	1.2	<b>2.3</b>	1.1	1.5	0.6
Task7-3	69.9	46.4	76.7	<b>81.0</b>	77.3	68.1	59.8
Task6-3	13.7	7.2	9.7	<b>10.4</b>	9.2	9.4	9.4
Task5-1	28.5	3.7	<b>6.0</b>	5.5	5.9	5.3	5.9
Task4-2	96.7	64.1	73.2	72.5	74.7	75.2	<b>76.7</b>
Task4-1	100.0	15.1	28.5	26.6	<b>36.2</b>	22.8	33.4
Task3-4	72.0	54.4	<b>79.9</b>	76.3	73.9	67.6	61.4
Task1-4	30.7	0.1	0.0	0.0	0.0	0.0	0.0

Table 3: Ablation results on ScienceWorld. “PANDA n” means learning from top-n responses generated from the expert model. “Raw 1” ablates PANDA insights to raw behavior of expert model (e.g., “the expert prefers A”). “Raw 2” ablates PANDA insights to raw preferences of expert model (e.g., “the expert prefers A rather than B”). We mark the best score except for the expert model in **bold** for each task.

examine the effectiveness of PANDA. As shown in Table 3, PANDA consistently outperforms the ablation cases of “Raw 1” or “Raw 2”, which consider raw preferences (e.g., “the expert prefers A rather than B”) without additional reasoning. This demonstrates that the learning process of PANDA, which involves gathering insights by reasoning about the preferences of the expert in specific contexts, leads to improved performance, except for only one exceptional case. We further explore the impact of incorporating preference data from the expert model by conducting ablation experiments on “PANDA n”, where n represents the learning process utilizing preference data of the top-n responses. We observed that “PANDA 2” achieves better performance than “PANDA 1”. Although the performance improvement of “PANDA 2” may appear to be marginal, our additional results on 20 tasks in Appendix B.1 further validate this conclusion. This indicates that learning from data that contains more comprehensive information about the preferences of the expert elicits more useful insights, which can effectively enhance the performance of language models. Ideally, having more preferences information from the expert would elicit more expert knowledge. However, our experimental results show no significant difference between “PANDA 3” and “PANDA 2”. This could be attributed to the difficulty for language models to reason and extract informative insights from multi-hop questions. Therefore, we decided to leverage “PANDA 2” for all our main experiments, as it demonstrated strong performance and captured valuable insights from the preferences data.

**Improvement derived from PANDA is not the substitution of more shots in the prompt.** We

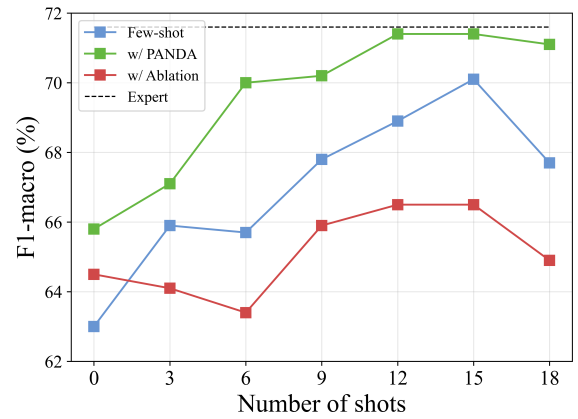


Figure 3: Ablation results on sentiment classification. For ablation study, we replace the retrieved insights with the corresponding few-shot examples (“w/ Ablation”), whose label is provided by the expert model.

conducted an additional experiment on the sentiment classification task to demonstrate that PANDA can surpass the performance ceiling of few-shot learning. In Fig. 3, we observe that the few-shot baseline achieves its highest performance when the number of shots is set to 15. Interestingly, PANDA consistently outperforms the few-shot baseline across all settings of shot numbers, indicating that improvement of PANDA cannot be substituted by adding more few-shot examples. To further investigate this, we performed an ablation where we replaced the retrieved insights with the corresponding few-shot examples (“w/ Ablation”), whose label is provided by the expert model. The results show that the ablation consistently performs worse than the baseline when the number of shots is greater than zero. This degradation in performance may be due to the pseudo labels generated by the expert model, as the expert model is not

perfect and its predictions may not always be accurate. In contrast, PANDA achieves consistent improvement by leveraging summary insights from the preferences of the expert. These insights better represent the expert’s understanding of the specific task compared to raw prediction results, leading to better generalization and improved performance.

## 5 Related Work

**Knowledge Distillation.** Knowledge Distillation (Hinton et al., 2015, KD) is a learning paradigm that involves training a inferior model to learn from a superior model in order to enhance the capabilities of the inferior model. Some work (Gu et al., 2023) proposes reverse KL divergence for generative language models to alleviating the problem caused by the distribution gap between teacher and student model. There is also work (Hsieh et al., 2023) leveraging LLMs to generate intermediate rationales that facilitates small language model to better handling the knowledge from large teacher model. Moreover, (Huang et al., 2022) implements Meta In-context Tuning and Multitask In-context Tuning to transfer few-shot learning ability of pre-trained language models to small language models. However, all of these methods are not only gradient-based that cannot be used on close-source models but also computation-intensive. As a contrast, our proposed PANDA leveraging the strong language understanding ability of LLMs is tuning-free. Meanwhile, most of works in KD leverage inferior models to learn from superior models, our work explores a new paradigm that leverages LLMs to learn from domain-specific expert models that may be much smaller than LLMs.

**Boosting LLMs with Small Language Models.** Lots of works leverage small language model to boost the performance of LLMs. Some methods focus on extending the task-specific capabilities of LLMs by utilizing small language models to enhance in-context examples (Xu et al., 2023) or by incorporating post-process modules (Vernikos et al., 2023). Retroformer (Yao et al., 2023) employs reinforcement learning to fine-tune a language model as a reflection-specific module, generating high-quality reflections in a loop that involves frozen LLMs. Another approach (Lu et al., 2023) employs small language models as decoding-time tailors for LLMs, aiming to improve task-specific performance. Some work (Liu et al., 2024) leverage the output distribution gap between pre- and

post-tuning of small language models as a heuristic to adjust the output distribution of LLMs, improving task-specific performance. Although PANDA also aims to leverage small models to enhance LLMs, there is a significant distinction that PANDA enables LLMs to learn from small models directly with a tuning-free manner, similar to the weak-to-strong paradigm proposed by OpenAI (Burns et al., 2023), thus being more flexible.

**Self-improvement of LLMs.** Building upon the strong foundational capabilities of LLMs, several approaches (Chen et al., 2024; Yuan et al., 2024; Qiao et al., 2024; Aksitov et al., 2023) have been proposed to enhance LLMs without the need for human supervision. To be specific, SPIN (Chen et al., 2024) treats LLMs as both opponents and main players to train LLMs using a bootstrapping way. Some work (Yuan et al., 2024) proposes to enable LLMs to generate both their training data and reward signals, allowing them to fine-tune themselves using the generated data. There are also works that tune language models with trajectory data generated by their own to boost themselves (Qiao et al., 2024; Aksitov et al., 2023). Moreover, weak-to-strong generalization proposed by (Burns et al., 2023) is another potential paradigm for boosting LLMs capabilities, which shares similarities with PANDA. A key difference is that PANDA leverages LLMs to learn from small domain-expert models in a purely tuning-free manner.

## 6 Conclusion

We propose PANDA, a tuning-free method that aims to improve the task-specific capability of LLMs. In PANDA, we leverage LLMs to learn from preferences of expert model and form an insight pool during learning stage. At inference time, PANDA firstly retrieved relevant insights to current query and boost the performance of LLMs by adapting its preference to align with the expert model with the guidance of relevant insights. Through extensive experiments on 20 tasks in ScienceWorld and 11 tasks in TweetEval, we verify that PANDA improve LLMs in most cases. Notably, PANDA achieves the weak-to-strong generalization on 4 tasks in ScienceWorld based on ReAct, which highlights the potential of exploring tuning-free approaches to achieve weak-to-strong generalization.



## Limitations

PANDA has two main limitations. Firstly, it relies on the retrieval process, where relevant insights need to be retrieved from the insight pool during inference. As the complexity of the task increases, retrieving the appropriate insights becomes more challenging, potentially causing the retrieval stage to become a performance bottleneck for PANDA. Secondly, PANDA is dependent on the powerful instruction-following capability of LLMs during the learning stage. This requirement restricts the usage of PANDA to LLMs with robust language understanding and instruction-following capabilities, limiting its applicability to open-sourced LLMs.

## References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. [Do as i can, not as i say: Grounding language in robotic affordances](#). *arXiv preprint arXiv:2204.01691*.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. 2023. [Rest meets react: Self-improvement for multi-step reasoning llm agent](#). *arXiv preprint arXiv:2312.10003*.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. [Tweeteval: Unified benchmark and comparative evaluation for tweet classification](#). *arXiv preprint arXiv:2010.12421*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *CoRR*, abs/2005.14165.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. [Weak-to-strong generalization: Eliciting strong capabilities with weak supervision](#). *arXiv preprint arXiv:2312.09390*.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. [Self-play fine-tuning converts weak language models to strong language models](#). *arXiv preprint arXiv:2401.01335*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. [Retrieval-augmented generation for large language models: A survey](#). *arXiv preprint arXiv:2312.10997*.

GeminiTeam, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. [Gemini: a family of highly capable multimodal models](#). *arXiv preprint arXiv:2312.11805*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. [Knowledge distillation of large language models](#). *arXiv preprint arXiv:2306.08543*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *arXiv preprint arXiv:1503.02531*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. [Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes](#). *arXiv preprint arXiv:2305.02301*.

Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. 2022. [In-context learning distillation: Transferring few-shot learning ability of pre-trained language models](#). *arXiv preprint arXiv:2212.10670*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. [Mixtral of experts](#). *arXiv preprint arXiv:2401.04088*.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniec, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. [Chatgpt: Jack of all trades, master of none](#). *Information Fusion*, page 101861.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. [Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning](#). *arXiv preprint arXiv:2304.05613*.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2023. [Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks](#). *arXiv preprint arXiv:2305.17390*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. [Tuning language models by proxy](#). *arXiv preprint arXiv:2401.08565*.

684	Ximing Lu, Faeze Brahman, Peter West, Jaehun Jang,	et al. 2022. <a href="#">Chain-of-thought prompting elicits rea-</a>	738
685	Khyathi Chandu, Abhilasha Ravichander, Lianhui	<a href="#">soning in large language models</a> . <i>Advances in Neural</i>	739
686	Qin, Prithviraj Ammanabrolu, Liwei Jiang, Sahana	<i>Information Processing Systems</i> , 35:24824–24837.	740
687	Ramnath, et al. 2023. <a href="#">Inference-time policy adapters</a>		
688	(ipa): Tailoring extreme-scale lms without fine-	Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu,	741
689	tuning. <i>arXiv preprint arXiv:2305.15065</i> .	Chenguang Zhu, and Julian McAuley. 2023. <a href="#">Small</a>	742
690	OpenAI. 2023. <a href="#">GPT-4 technical report</a> . <i>CoRR</i> ,	<a href="#">models are valuable plug-ins for large language mod-</a>	743
691	abs/2303.08774.	<a href="#">els</a> . <i>arXiv preprint arXiv:2305.08848</i> .	744
692	Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021.	Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu,	745
693	<a href="#">Dialogue graph modeling for conversational machine</a>	Quoc V Le, Denny Zhou, and Xinyun Chen. 2023.	746
694	<a href="#">reading</a> . In <i>Findings of the Association for Com-</i>	<a href="#">Large language models as optimizers</a> . <i>arXiv preprint</i>	747
695	<i>putational Linguistics: ACL-IJCNLP 2021</i> , pages	<i>arXiv:2309.03409</i> .	748
696	3158–3169, Online. Association for Computational		
697	Linguistics.	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak	749
698	Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi	Shafran, Karthik Narasimhan, and Yuan Cao. 2022.	750
699	Jia, Prateek Mittal, and Peter Henderson. 2023. <a href="#">Fine-</a>	<a href="#">React: Synergizing reasoning and acting in language</a>	751
700	<a href="#">tuning aligned language models compromises safety,</a>	<a href="#">models</a> . <i>arXiv preprint arXiv:2210.03629</i> .	752
701	<a href="#">even when users do not intend to!</a> <i>arXiv preprint</i>	Weiran Yao, Shelby Heinecke, Juan Carlos Niebles,	753
702	<i>arXiv:2310.03693</i> .	Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy,	754
703	Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo,	Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al.	755
704	Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei	2023. <a href="#">Retroformer: Retrospective large language</a>	756
705	Lv, and Huajun Chen. 2024. <a href="#">Autoact: Automatic</a>	<a href="#">agents with policy gradient optimization</a> . <i>arXiv</i>	757
706	<a href="#">agent learning from scratch via self-planning</a> . <i>arXiv</i>	<i>preprint arXiv:2308.02151</i> .	758
707	<i>preprint arXiv:2401.05268</i> .	Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho,	759
708	Nils Reimers and Iryna Gurevych. 2019. <a href="#">Sentence-bert:</a>	Sainbayar Sukhbaatar, Jing Xu, and Jason Weston.	760
709	<a href="#">Sentence embeddings using siamese bert-networks</a> .	2024. <a href="#">Self-rewarding language models</a> . <i>arXiv</i>	761
710	<i>arXiv preprint arXiv:1908.10084</i> .	<i>preprint arXiv:2401.10020</i> .	762
711	Noah Shinn, Federico Cassano, Ashwin Gopinath,	Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and	763
712	Karthik R Narasimhan, and Shunyu Yao. 2023. <a href="#">Re-</a>	Dacheng Tao. 2023. <a href="#">Can chatgpt understand too?</a>	764
713	<a href="#">flexion: Language agents with verbal reinforcement</a>	<a href="#">a comparative study on chatgpt and fine-tuned bert</a> .	765
714	<a href="#">learning</a> . In <i>Thirty-seventh Conference on Neural</i>	<i>arXiv preprint arXiv:2302.10198</i> .	766
715	<i>Information Processing Systems</i> .	Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B	767
716	Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-	Brown, Alec Radford, Dario Amodei, Paul Chris-	768
717	bert, Amjad Almahairi, Yasmine Babaei, Nikolay	tiano, and Geoffrey Irving. 2019. <a href="#">Fine-tuning lan-</a>	769
718	Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti	<a href="#">guage models from human preferences</a> . <i>arXiv</i>	770
719	Bhosale, et al. 2023. <a href="#">Llama 2: Open founda-</a>	<i>preprint arXiv:1909.08593</i> .	771
720	<a href="#">tion and fine-tuned chat models</a> . <i>arXiv preprint</i>		
721	<i>arXiv:2307.09288</i> .		
722	Giorgos Vernikos, Arthur Bražinskas, Jakub Adamek,		
723	Jonathan Mallinson, Aliaksei Severyn, and Eric		
724	Malmi. 2023. <a href="#">Small language models improve</a>		
725	<a href="#">giants by rewriting their outputs</a> . <i>arXiv preprint</i>		
726	<i>arXiv:2305.13514</i> .		
727	Danqing Wang and Lei Li. 2023. <a href="#">Learning from mis-</a>		
728	<a href="#">takes via cooperative study assistant for large lan-</a>		
729	<a href="#">guage models</a> . In <i>Proceedings of the 2023 Confer-</i>		
730	<i>ence on Empirical Methods in Natural Language</i>		
731	<i>Processing</i> , pages 10667–10685.		
732	Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and		
733	Prithviraj Ammanabrolu. 2022. <a href="#">Scienceworld: Is</a>		
734	<a href="#">your agent smarter than a 5th grader?</a> <i>arXiv preprint</i>		
735	<i>arXiv:2203.07540</i> .		
736	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten		
737	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,		

## A Experimental Setup

In all experiments, we take gpt-3.5-turbo-1106 as the LLM in PANDA. Due to the observed variability in results within the ScienceWorld benchmark, even when the temperature is fixed at 0.0, we conducted each experiment for 5 rounds and recorded the average score as the final measurement.

### A.1 ScienceWorld

#### A.1.1 Dataset Statistics

To save time and cost while conducting extensive experiments in ScienceWorld, we adopted the same settings as (Lin et al., 2023). Without loss of generality, we randomly select 2 tasks from each task type, resulting in a total of 20 tasks. For tasks with more than 50 variations in the training set, we randomly sampled 50 variations as the training set. Additionally, we randomly sampled up to 10 variations from the test set for performance evaluation. In Table 5, we provide detailed dataset statistics for ScienceWorld based on the aforementioned sampling approach.

#### A.1.2 Prompt Template

In this section, we present the prompt templates when implementing PANDA in ScienceWorld in Table 6 and 7.

### A.2 TweetEval

#### A.2.1 Dataset Statistics

Due to time and cost constraints, and without sacrificing generalization, we randomly sampled up to 1000 samples as the test data. As our goal with PANDA is to be a sample-efficient method, we use the same size for the training data. Detailed statistics for the dataset we use are presented in Table 8.

#### A.2.2 Prompt Template

In this section, we provide the prompt templates when implementing PANDA in TweetEval. We present prompt templates for both PANDA-Learning, PANDA-Inference and baselines in Fig. 9 to 12.

## B Additional Experimental Results

### B.1 ScienceWorld

We conduct additional experiments to validate the effectiveness of learning from preferences compared to learning from individual behavior. As

shown in Table 4, PANDA with learning from preference pair outperforms PANDA with learning from individual behavior.

818  
819  
820

Task	Expert	ReAct	ReAct <sub>p2</sub>	ReAct <sub>p1</sub>
Task10-2	17.0	0.2	2.1	<b>6.4</b>
Task10-1	16.9	21.7	38.9	<b>43.5</b>
Task9-3	66.5	10.0	<b>12.8</b>	12.1
Task9-2	41.5	16.5	21.7	<b>23.3</b>
Task8-2	36.6	0.0	<b>2.3</b>	1.1
Task8-1	8.0	4.0	<b>7.0</b>	6.0
Task7-3	69.9	46.4	<b>81.0</b>	77.3
Task7-1	85.0	50.0	<b>95.5</b>	91.5
Task6-3	13.7	7.2	<b>10.4</b>	9.2
Task6-1	22.9	18.0	<b>26.4</b>	21.7
Task5-2	17.0	<b>36.2</b>	23.7	22.3
Task5-1	28.5	3.7	5.5	<b>5.9</b>
Task4-2	96.7	64.1	72.5	<b>74.7</b>
Task4-1	100.0	15.1	26.6	<b>36.2</b>
Task3-4	72.0	64.8	<b>76.3</b>	73.9
Task3-1	73.8	15.4	22.8	<b>32.9</b>
Task2-3	5.8	33.3	<b>37.7</b>	17.0
Task2-1	8.7	<b>11.3</b>	9.6	7.0
Task1-4	30.7	<b>0.1</b>	0.0	0.0
Task1-1	44.8	0.0	<b>0.2</b>	0.1

Table 4: Ablation results on ScienceWorld. ReAct<sub>p2</sub> and ReAct<sub>p1</sub> represents PANDA learning from preference pair and individual behavior respectively. We mark the best score except for the expert model in **bold** for each task.

Task Type	Topic	Name	#Vars: Train	#Vars: Test	# Actions
Task10-2	Biology	mendelian-genetics-unknown-plant	50	10	1370
Task10-1	Biology	mendelian-genetics-known-plant	50	10	1377
Task9-3	Forces	inclined-plane-friction-unnamed-surfaces	50	10	382
Task9-2	Forces	inclined-plane-friction-named-surfaces	50	10	395
Task8-2	Biology	identify-life-stages-1 (animal)	6	5	37
Task8-1	Biology	identify-life-stages-2 (plant)	4	4	109
Task7-3	Biology	lifespan-longest-lived-then-shortest-lived	50	10	228
Task7-1	Biology	lifespan-longest-lived	50	10	205
Task6-3	Chemistry	chemistry-mix-paint-tertiary-color	18	9	325
Task6-1	Chemistry	chemistry-mix	16	8	136
Task5-2	Biology	grow-fruit	50	10	2137
Task5-1	Biology	grow-plant	50	10	2728
Task4-2	Classification	find-non-living-thing	50	10	278
Task4-1	Classification	find-living-thing	50	10	1013
Task3-4	Electricity	test-conductivity-of-unknown-substances	50	10	1437
Task3-1	Electricity	power-component (circuit)	10	5	189
Task2-3	Measurement	measure-melting-point-unknown-substance	50	10	285
Task2-1	Measurement	use-thermometer	50	10	227
Task1-4	Matter	change-the-state-of-matter-of	14	9	233
Task1-1	Matter	boil	14	9	273

Table 5: The statistics of ScienceWorld benchmark we use. We present the variations number of train-test splits for each task and the number of resulted actions in the training data.

Prompt Template of PANDA-Inference for ScienceWorld	
{Init Prompt}	
These are some insights that may be helpful for you to improve the success rate:	
{Retrieved Insights}	
{Current Trajectory}	

Table 6: Prompt template of PANDA-Inference for ScienceWorld.



---

**Prompt Template of PANDA-Learning for ScienceWorld**

{Current Trajectory}

Now it's time to act again, the expert prefers {most preferred action} rather than {second preferred action}. Please explain the reason why the expert holds on this preference.

---

**Example of PANDA-Learning in ScienceWorld**

#prompt here#

The expert's trial up to now is as follows:

Here is the task.

This room is called the kitchen. In it, you see: the agent a substance called air a chair. On the chair is: nothing. a counter. On the counter is: a bowl (containing a red apple, a banana, an orange, a potato), a drawer. a cupboard. The cupboard door is closed. a freezer. The freezer door is closed. a fridge. The fridge door is closed. a glass jar (containing a substance called sodium chloride) a lighter a oven, which is turned off. The oven door is closed. a painting a sink, which is turned off. In the sink is: nothing. a substance called soap a stopwatch, which is deactivated. a stove, which is turned off. On the stove is: nothing. a table. On the table is: a glass cup (containing nothing). a thermometer, currently reading a temperature of 10 degrees celsius You also see: A door to the bathroom (that is open) A door to the hallway (that is open) A door to the outside (that is open) Your task is to find the animal with the longest life span, then the shortest life span. First, focus on the animal with the longest life span. Then, focus on the animal with the shortest life span. The animals are in the 'outside' location.

> go to outside

You move to the outside.; In your inventory, you see: an orange ; This outside location is called the outside. Here you see: the agent a substance called air an axe a chameleon egg a fire pit (containing nothing) a fountain (containing a substance called water) a giant tortoise egg the ground a baby rabbit a substance called wood You also see: A door to the foundry (that is open) A door to the greenhouse (that is open) A door to the kitchen (that is open)

>

Now it's time to act again, the expert prefers to focus on egg giant tortoise rather than to focus on chameleon. Please explain the reason why the expert holds on this preference.

Expert\_insight:

#response here#

INSIGHT: The expert prefers to focus on the giant tortoise egg rather than the chameleon egg because the giant tortoise is known to have a significantly longer lifespan compared to the chameleon.

Giant tortoises are known to live for over 100 years, with some species even reaching ages of 150 years or more. On the other hand, chameleons have a relatively shorter lifespan, typically ranging from 2 to 10 years depending on the species. Therefore, the expert believes that investigating the giant tortoise egg would be more likely to yield information about the animal with the longest lifespan.

---

Table 7: Prompt template and an example of PANDA-Learning for ScienceWorld.

Task	# Train Data	# Test Data	# Class
<b>Emoji</b>	1000	1000	20
<b>Emotion</b>	1000	1000	4
<b>Offensive</b>	860	860	2
<b>Sentiment</b>	1000	1000	3
<b>Hate</b>	1000	1000	2
<b>Irony</b>	784	784	2
<b>Stance-Abortion</b>	280	280	3
<b>Stance-Atheism</b>	220	220	3
<b>Stance-Climate</b>	169	169	3
<b>Stance-Feminist</b>	285	285	3
<b>Stance-Hillary</b>	295	295	3

Table 8: Dataset statistics of TweetEval we use.

---

<b>Prompt Template of PANDA-Inference for TweetEval</b>
These are some insights that may be helpful for you to improve the success rate:
{Retrieved Insights}
{zero-shot/few-shot/cot prompt}

---

Table 9: Prompt template of PANDA-Inference for TweetEval.

---

**Prompt Template of PANDA-Learning for TweetEval**

#prompt here#

Output the {task name} of the given text.

Choose your answer from provided list and map your answer with following {mapping in task} and return an integer as a result.

{Text}

Answer ({candidate answer}):

The expert prefers {most preferred answer} rather than {second preferred answer}.

Please explain the reason why the expert holds on this preference.

To determine the {task name} of a given text,

#response here#

INSIGHT:

---

**Example of PANDA-Learning in TweetEval**

#prompt here#

Output the sentiment of the given text. Choose your answer from provided list and map your answer with following negative: 0, neutral: 1, positive: 2 and return an integer as a result.

Text: Today is Amazon Prime Day. Today is the best day to buy. July 15th only. #savemoney

Answer (0 or 1 or 2):

The expert prefers positive(2) rather than neutral(1).

Please explain the reason why the expert holds on this preference.

To determine the sentiment of a given text,

#response here#

INSIGHT: To determine the sentiment of a given text,

the expert looks for words and phrases that convey a positive or negative tone.

In this text, the use of words like "best day" and "save money" convey a positive sentiment, indicating that today is a good day to make purchases and take advantage of deals.

Therefore, the sentiment of the given text is positive (2).

---

Table 10: Prompt template and an example of PANDA-Learning.

---

**Prompt Template of Zero-shot for TweetEval**

Output the {task name} of the given text.

Choose your answer from provided list and map your answer with following {mapping in task} and return an integer as a result.

{Text}

Answer ({candidate answer}):

---

**Prompt Template of Few-shot for TweetEval**

Output the sentiment of the given text. Choose your answer from provided list and map your answer with following negative: 0, neutral: 1, positive: 2 and return an integer as a result.

Text: Dark Souls 3 April Launch Date Confirmed With New Trailer: Embrace the darkness.

Answer (0 or 1 or 2): 1

Output the sentiment of the given text. Choose your answer from provided list and map your answer with following negative: 0, neutral: 1, positive: 2 and return an integer as a result.

Text: "National hot dog day, national tequila day, then national dance day... Sounds like a Friday night."

Answer (0 or 1 or 2): 2

Output the sentiment of the given text. Choose your answer from provided list and map your answer with following negative: 0, neutral: 1, positive: 2 and return an integer as a result.

Text: When girls become bandwagon fans of the Packers because of Harry.

Do y'all even know who Aaron Rodgers is? Or what a 1st down is?

Answer (0 or 1 or 2): 0

Output the {task name} of the given text.

Choose your answer from provided list and map your answer with following {mapping in task} and return an integer as a result.

{Text}

Answer ({candidate answer}):

---

Table 11: Prompt template of Zero-shot and Few-shot for TweetEval.



---

**Prompt Template of Zero-shot CoT for TweetEval**

Output the {task name} of the given text.

Choose your answer from provided list and map your answer with following {mapping in task} and return an integer as a result. Before output the answer, think it step by step firstly.

{Text}

Let's think step by step (end your answer with "the answer is #integer#"):

---

**Prompt Template of Few-shot CoT for TweetEval**

Output the sentiment of the given text. Choose your answer from provided list

and map your answer with following negative: 0, neutral: 1, positive: 2

and return an integer as a result. Before output the answer, think it step by step firstly.

Text: Dark Souls 3 April Launch Date Confirmed With New Trailer: Embrace the darkness.

Let's think step by step (end your answer with "the answer is #integer#"):

1. The text mentions the launch date being confirmed for Dark Souls 3, which is likely to be exciting news for fans of the game. This could indicate a positive sentiment.

2. The phrase "Embrace the darkness" could be interpreted as a call to action or a thematic element of the game, which may not necessarily indicate a negative sentiment.

3. Overall, the text seems to convey a sense of anticipation and excitement for the upcoming release.

Based on this analysis, the sentiment of the given text can be categorized as neutral.

Therefore, the answer is 1.

Output the sentiment of the given text. Choose your answer from provided list

and map your answer with following negative: 0, neutral: 1, positive: 2

and return an integer as a result. Before output the answer, think it step by step firstly.

Text: "National hot dog day, national tequila day, then national dance day... Sounds like a Friday night."

Let's think step by step (end your answer with "the answer is #integer#"):

Step 1: Identify key words and phrases

- "National hot dog day" - "national tequila day" - "national dance day" - "Sounds like a Friday night"

Step 2: Analyze the tone and context

- The text seems to be expressing excitement or anticipation for the events mentioned.

- The mention of "Friday night" suggests a positive and fun atmosphere.

Step 3: Determine the sentiment - The overall sentiment of the text is positive.

Step 4: Map the sentiment to the provided list - Positive: 2

So, the answer is 2.

Output the sentiment of the given text. Choose your answer from provided list

and map your answer with following negative: 0, neutral: 1, positive: 2 and

return an integer as a result. Before output the answer, think it step by step firstly.

Text: When girls become bandwagon fans of the Packers because of Harry.

Do y'all even know who Aaron Rodgers is? Or what a 1st down is?

Let's think step by step (end your answer with "the answer is #integer#"):

Step 1: The text seems to be mocking girls who become fans of the Packers because of Harry, implying that they may not actually know much about football.

Step 2: The tone of the text is somewhat negative and condescending.

Step 3: Based on the above analysis, the sentiment of the text is negative.

Step 4: Mapping the sentiment "negative" to the provided list, we get negative: 0.

Step 5: Therefore, the integer result for the sentiment of the text is 0.

So, the answer is 0.

Output the {task name} of the given text.

Choose your answer from provided list and map your answer with following {mapping in task} and return an integer as a result. Before output the answer, think it step by step firstly.

{Text}

Let's think step by step (end your answer with "the answer is #integer#"):

---

Table 12: Prompt template of Zero-shot and Few-shot CoT for TweetEval.