Agent-FLAN: Designing Data and Methods of Effective Agent Tuning for Large Language Models

Zehui Chen [1] Kuikun Liu [2] Qiuchen Wang [1] Wenwei Zhang [2] Jiangning Liu [2]
Dahua Lin [2] Kai Chen [2] Feng Zhao [1]

Abstract

Open-sourced Large Language Models (LLMs) have achieved great success in various NLP tasks, however, they are still far inferior to API-based models when acting as agents. How to integrate agent ability into general LLMs becomes a crucial and urgent problem. This paper first delivers three key observations: (1) the current agent training corpus is entangled with both formats following and agent reasoning, which significantly shifts from the distribution of its pre-training data; (2) LLMs exhibit different learning speeds on the capabilities required by agent tasks; and (3) current approaches have side-effects when improving agent abilities by introducing hallucinations. Based on the above findings, we propose Agent-FLAN to effectively Fine-tune LANguage models for Agents. Through careful decomposition and redesign of the training corpus, Agent-FLAN enables Llama2-7B to outperform prior best works by 3.5% across various agent evaluation datasets. With comprehensively constructed negative samples, Agent-FLAN greatly alleviates the hallucination issues based on our established evaluation benchmark. Besides, it consistently improves the agent capability of LLMs when scaling model sizes while slightly enhancing the general capability of LLMs. The code will be available at https://github.com/InternLM/Agent-FLAN.

1. Introduction

Language agent (Mialon et al., 2023; Sumers et al., 2023), which leverages the powerful abilities of LLMs to perceive environments, make decisions, and take actions, has emerged as an effective solution to complicated real-world problems (Song et al., 2023; Huang et al., 2023). Existing studies mainly focus on either prompt engineering (Wei et al., 2022; Yao et al., 2022b; Shinn et al., 2023) or framework scheduling of one/multiple close-sourced LLM (GPT-4) (Wu et al., 2023; Talebirad & Nadiri, 2023; Hong et al., 2023) to accomplish agent tasks. Despite the impressive results and flexibility of these works, close-sourced LLMs suffer from steep financial cost and safety issues (Li et al., 2023; Yuan et al., 2023), impeding them from further deployment.

Recently, open-sourced LLMs emerged as effective alternatives and show promising results in various applications (Chiang et al., 2023; Touvron et al., 2023; Jiang et al., 2024). However, a lingering challenge persists: while these LLMs excel in tasks necessitating linguistic finesse, they fall short when deployed as agents, especially when compared to API-based models (Zeng et al., 2023; Liu et al., 2023a). Efforts have been made to fine-tune LLMs for specific vertical fields (Qin et al., 2023; Gou et al., 2023), but this approach risks compromising their general capabilities. Thus, integrating effective agent abilities into general LLMs becomes a critical and pressing challenge in the field, aiming to bridge the gap between specialized and versatile language models.

In this paper, we embark on a journey to mitigate this disparity between the prowess of open-sourced LLMs and the superior performance exhibited by API-based models in agent-centric tasks. We first unravel three critical observations that serve as the foundation for our exploration: (1) most agent training data is entangled with both format following and general reasoning, deviating from the original pretrain domain of the model, *i.e.,* natural conversation, leading to inadequate agent learning, (2) by explicitly decomposing the training data along the basic capabilities, each loss exhibits different convergence curves, indicating varied learning speeds on the abilities requisite for agent tasks, and (3) the existing approaches that predominantly concentrate on specific agent abilities, overlooking the prevalence and significance of hallucinations in the model's output.

Building upon these key insights, we present Agent-FLAN, which investigates the designing choice of effective **Agent**

[1]Department of Automation, University of Science and Technology of China [2]Shanghai AI Labortary. Correspondence to: Kai Chen, Feng Zhao <chenkai@pjlab.org.cn, fzhao956@ustc.edu.cn>.

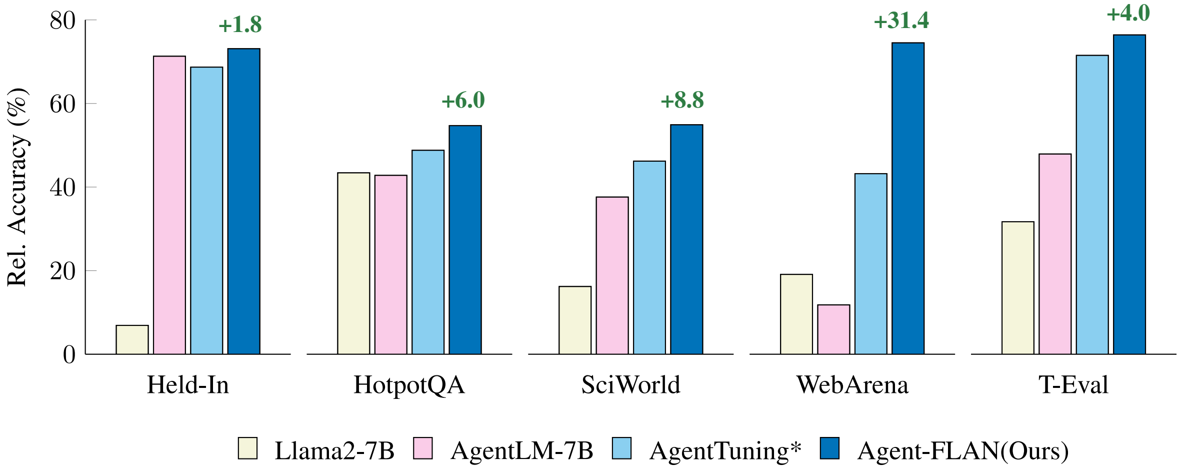


Figure 1. **Comparison of recent agent tuning approaches on Held-In, Held-Out tasks**. Performances are normalized with GPT-4 results for better visualization. * denotes our re-implementation for a fair comparison.

Finetuning for large **LAN**guage models. Specifically, we tease apart format following and common reasoning in the agent training corpus, aligning the fine-tuning process to the pretrained domain of the language model. This elicits the pure agent abilities in LLMs without overfitting to specific format protocols. By further disentangling the agent tasks into distinct facets along the fundamental competencies of LLMs, Agent-FLAN affords training flexibilities depending on the varying learning rates of respective competencies. To comprehensively resolve the hallucination issues in agent tasks, we construct the *Agent-H* benchmark, which assesses the hallucination issues of LLMs from various aspects. Subsequently, we meticulously curate diverse 'negative' training samples to mitigate this problem effectively.

We employ Agent-FLAN on the open-sourced Llama2-series, which surpasses prior works by a substantial 3.5% margin across a spectrum of agent evaluation benchmarks, including general agent tasks and tool utilization. Furthermore, we provide a deeper understanding of the dynamics involved in agent tuning: the scaling laws governing data and model dimensions, and the intricate relationships between general and agent-specific tasks. Our major contributions are as follows:

- We identify three critical observations that hinder open-sourced LLMs from achieving competitive performance in the agent domain, offering valuable insights into the complex landscape of agent tuning.
- Based on above findings, we introduce Agent-FLAN, an innovative approach aiming to integrate effective agent abilities into general LLMs: aligning agent tuning to chat format (§4.1), capabilities decomposition and data balancing (§4.2), and negative sample constructions for hallucination elimination (§4.3).
- Agent-FLAN outperforms prior works by a substantial 3.5% margin on Llama2-series across a spectrum of agent evaluation benchmarks. Besides, we further study the dynamics of agent tuning, including scaling laws in terms of data and model scales, and intricate relationships between general and agent-specific tasks.

2. Related Work

2.1. LLM as Agent

The emergence of large language models (LLMs) represents a significant stride towards Artificial General Intelligence (AGI). With the advancement of LLMs, language agents built upon LLMs engage with the world to accomplish a diverse array of tasks, having become a focal point of research (Wang et al., 2023a; Xi et al., 2023). LLMs have been utilized in various studies for specific agent tasks, including web browsing(Deng et al., 2023; Zhou et al., 2023), online shopping (Yao et al., 2022a), database operations(Liu et al., 2023a), science experiments (Wang et al., 2022a), observation free reasoning (Xu et al., 2023), Wikipedia Q&A (Yang et al., 2018), daily computer tasks (Kim et al., 2023) and household exploration (Shridhar et al., 2020).

In addition to research focused on specific tasks, there are ongoing studies concerning AI agents based on LLMs. Some studies, such as ReAct(Yao et al., 2022b), emphasize actions during thinking, leading to significant improvements over various methods. While other works focus primarily on human and social property inside agents (Mao et al., 2023; Park et al., 2023; Zhang et al., 2023a), intelligence collaboration within multiple agents (Chen et al., 2023b; Liu et al., 2023b; Liang et al., 2023). Different from above, Agent-FLAN facilitates the integration of effective agent

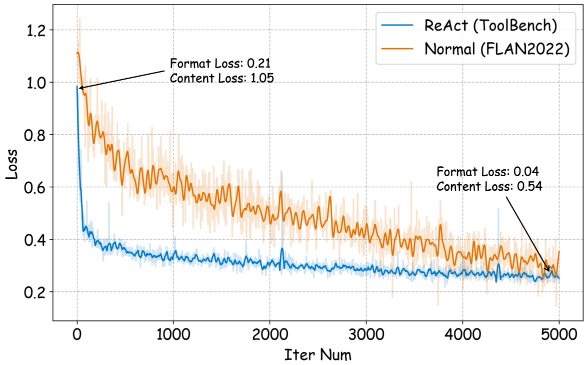


Figure 2. Comparison of training loss on ReAct data (Toolbench (Qin et al., 2023)) and normal conversation (Flan2022 (Longpre et al., 2023)).

capabilities into general LLMs, enabling models to better understand and tackle complex problems in the real world.

2.2. Language Model Fine-Tuning

Language Model Fine-Tuning is also a research hotspot, involving the adjustment of pre-trained models to adapt to specific tasks, aiming to align the output with expectations (Zhang et al., 2023b). Various research studies have been conducted on fine-tuning to optimize the model's reasoning capabilities (Liu et al., 2021; Fu et al., 2023), proficiency in tools (Patil et al., 2023; Qin et al., 2023; Qiao et al., 2023), planning capability (Chen et al., 2023a), retrieval-augmented (Wang et al., 2023b), etc. Additionally, there are various studies conducted on fine-tuning methods (Hu et al., 2021; Ivison et al., 2022; Dettmers et al., 2023; Lv et al., 2023), data selection principles (Gunasekar et al., 2023) and fine-tuning datasets (Sanh et al., 2021; Wang et al., 2022b; Honovich et al., 2022; Longpre et al., 2023).

3. Pilot Observations

In this section, we delve into three pivotal observations on agent tuning that serve as the foundation of our subsequent investigation.

Observation 1. *Most agent training data is entangled with both format following and general reasoning, causing a significant departure from the model's original pretraining language domain, namely, natural conversation.*

Recent agent tuning works (Zeng et al., 2023; Qin et al., 2023) endorse the adoption of specific formats, exemplified by ReAct (Thought-Action-Observation), for fine-tuning the language model. Moreover, it's noteworthy that action arguments are frequently presented in JSON format. Encoding both format and reasoning knowledge into the training corpus shifts the tuning process from the original chat domain, presenting it as an out-of-domain task for language models. As shown in Figure 2, we compare the training curve of formatted data and normal data. It can be clearly seen that the loss associated with formatted data descends more rapidly to a low value, while keeping content loss still high (0.54 vs 0.04), indicating that the former leads to an inadequate learning process. This phenomenon can be probably attributed to the existence of a fixed structure (ReAct, JSON), where the model quickly gets overfitted to the format itself. Consequently, it fails to grasp the underlying reasoning abilities embedded within the training data, resulting in unsatisfied performance.

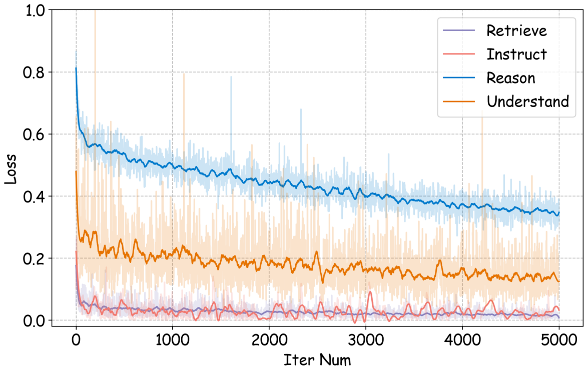


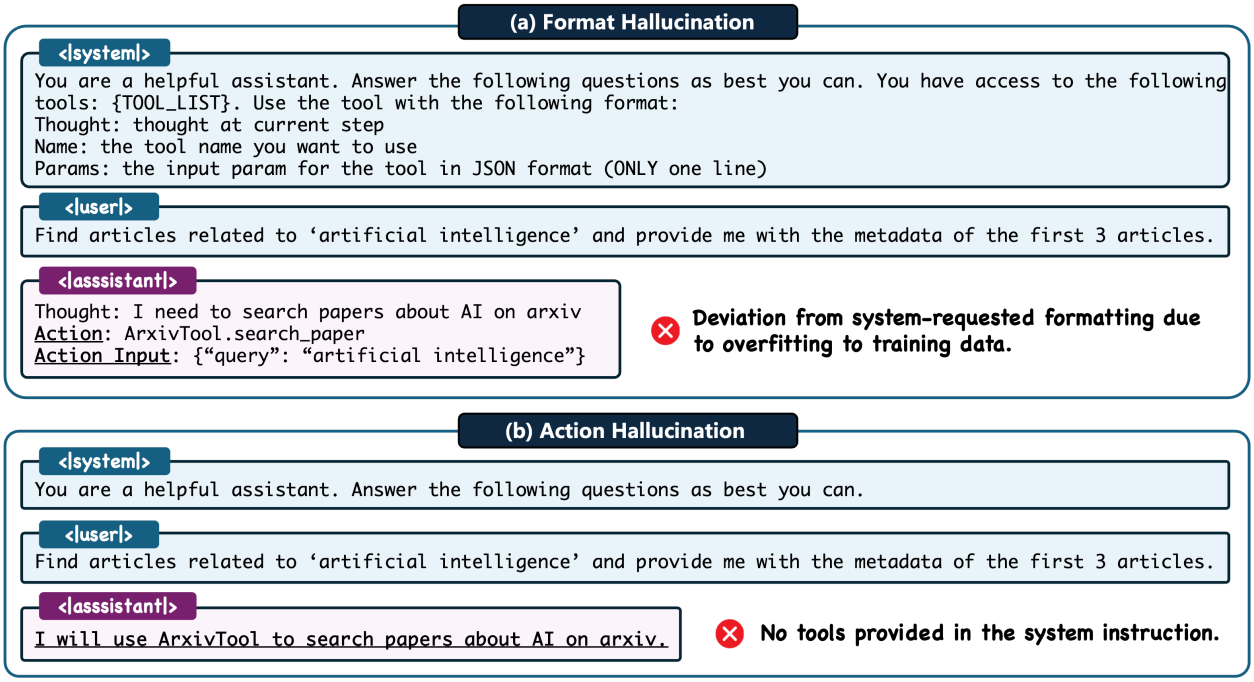
Figure 3. Visualization of training loss by decomposing it into different capabilities of models: retrieval, instruct following, reasoning, and understanding.

Observation 2. *By explicitly decomposing the training data along the basic capability aspects, each loss exhibits different convergence curves, indicating varied learning speeds on the capabilities requisite for agent tasks of LLMs.*

Inspired by (Chen et al., 2023c), we explicitly disentangle the model's capabilities into distinct components: instruction following, reasoning, retrieval, and understanding. In this context, instruction following corresponds to format generation, reasoning corresponds to the thought quality at each step, retrieval involves selecting the appropriate function name to execute the task, and the understanding encompasses the parameter inputs for the selected functions. By visualizing the loss based on respective aspects in Figure 3, we discern that LLM tends to exhibit varying learning speeds for the capabilities essential for proficient agents. To elaborate, retrieval and understanding emerge as relatively more manageable tasks compared to reasoning, with instruction following being the simplest in the learning process. This observation serves as a compelling motivation to further disentangle the training data along these model capabilities and subsequently balance these data based on the model's varying learning rates.

Observation 3. *Existing approaches predominantly concentrate on specialized agent abilities, overlooking the prevalence and significance of hallucination effects in the model's output.*

AgentTuning (Zeng et al., 2023) introduces mixture train-



```
You are a helpful assistant. Answer the following questions as best you can. You have access to the following
tools: {TOOL_LIST}. Use the tool with the following format:
Thought: thought at current step
Name: the tool name you want to use
Params: the input param for the tool in JSON format (ONLY one line)
```

```
Find articles related to ‘artificial intelligence’ and provide me with the metadata of the first 3 articles.
```

```
Thought: I need to search papers about AI on arxiv
Action: ArxivTool.search_paper
Action Input: {“query”: “artificial intelligence”}
```

Deviation from system-requested formatting due to overfitting to training data.
(b) Action Hallucination
<|system|>

```
You are a helpful assistant. Answer the following questions as best you can.
```

```
Find articles related to ‘artificial intelligence’ and provide me with the metadata of the first 3 articles.
```

```
I will use ArxivTool to search papers about AI on arxiv.
```

Figure 4. Illustration of two typical hallucinations in general agent tasks for current open-sourced LLMs: (a) format hallucination and (b) action hallucination.

ing, where both a general dataset and an agent dataset are simultaneously provided to the model during the tuning process. While this strategy indeed leads to steady performance improvements, we observe that it has limited impact on addressing hallucination issues, *i.e.,* a crucial concern often overlooked in recent agent studies. This concern becomes particularly significant when deploying language models in real-world applications, as illustrated in Figure 4. Hallucination manifests in two primary aspects: (1) When the model is required to invoke a response, it strictly adheres to the training format, disregarding the user-generated query, and (2) The model is susceptible to being triggered with non-existent functions when presented with induced questions. This underscores the necessity of directing more attention toward refining agent tuning mechanisms and the establishment of appropriate benchmarks to assess and mitigate agent hallucination effectively.

4. Agent-FLAN

Recent works start to explore the effectiveness of finetuning language models on various agent tasks, with varied data quality, model sizes, and tuning approaches. Building on the pilot observations, we dive deeper into the data and method designing choice of effective agent tuning and discuss three key improvements to the language agent models.

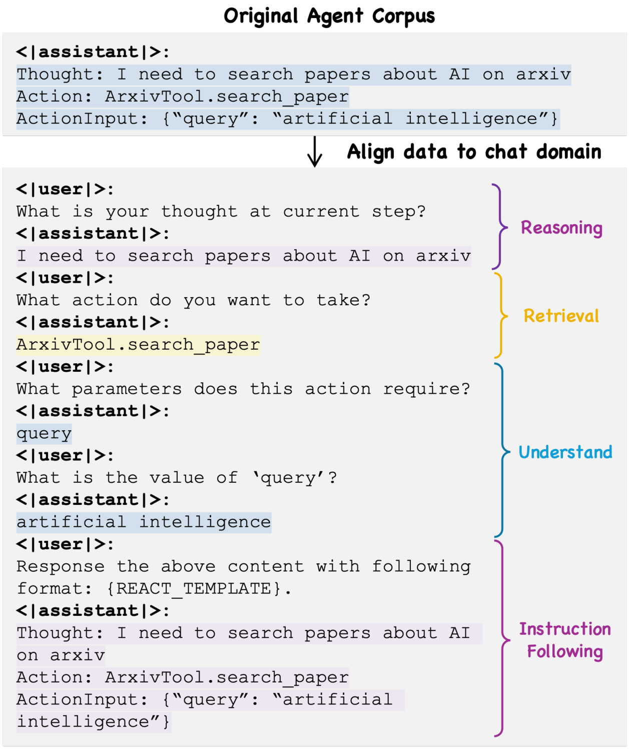
Experimental Setup We finetune the language model Llama2-series (Touvron et al., 2023), and use 7B size for ablations for efficiency unless otherwise stated. We construct the dataset in accordance with the data and settings established by AgentTuning (Zeng et al., 2023). Specifically, we pick a suite of training data sources as held-in tasks: ALFWorld (Shridhar et al., 2020), WebShop (Yao et al., 2022a), Mind2Web (Deng et al., 2023), Knowledge Graph (Liu et al., 2023a), Operating System (Liu et al., 2023a), Database (Liu et al., 2023a), and ToolBench (Qin et al., 2023), covering both general agents and tool utilization domains. Our held-out evaluation encompasses a list of complex interactive tasks, including complex QA (HotpotQA (Yang et al., 2018)), web browsing (WebArena (Zhou et al., 2023)), science experiments (SciWorld (Wang et al., 2022a)), and tool utilization (T-Eval (Chen et al., 2023c)). Details adopted in Agent-FLAN and hyper-parameters during training can be found in Appendix A.

4.1. Aligning Agent Tuning to Pretrain Domain

LLMs are first pretrained on natural language corpus, which comprises a massive number of human conversations. However, agent data are often presented in specific formats (ReAct, JSON), leading to out-of-distribution learning during the tuning process. Such a misalignment further results in an inadequate learning process. Besides, LLMs are more likely to get overfitted to these specific formats after finetuning, deteriorating their instruction-following abilities. To mitigate this problem, we propose to transform the formatted data into natural conversations. Concretely, we first

Table 1. **Main results of Agent-FLAN**. Agent-FLAN significantly outperforms previous agent-tuning approaches by a large margin on both held-in and held-out tasks. * denotes our re-implementation with the same amount of training data for a fair comparison. Since FireAct does not train on AgentInstruct dataset, we omit its performance on the HELD-IN set. **Bold**: the best in API-based and open-sourced models.

MODEL	HELD-IN	HELD-OUT					
		HotpotQA	SciWorld	WebArena	T-Eval	Agent-H	Overall
GPT-3.5 (OpenAI, 2022)	1.59	37.4	21.2	4.56	84.0	92.1	47.8
GPT-4 (OpenAI, 2023)	**2.75**	**52.1**	**36.4**	**6.28**	**86.4**	**94.2**	**55.1**
Llama2-7B (Touvron et al., 2023)	0.19	22.6	5.9	1.2	27.4	78.7	27.2
FireAct-7B (Chen et al., 2023a)	-	26.2	6.8	0.25	9.3	40.4	16.6
AgentLM-7B (Zeng et al., 2023)	1.96	22.3	13.7	0.74	41.4	80.6	31.7
AgentTuning* (Zeng et al., 2023)	1.89	25.4	16.8	2.71	61.8	84.5	38.2
Agent-FLAN (Ours)	**2.01**	**28.5**	**20.0**	**4.68**	**66.0**	**89.1**	**41.7**



Original Agent Corpus

```
<|assistant|>:
Thought: I need to search papers about AI on arxiv
Action: ArxivTool.search_paper
ActionInput: {"query": "artificial intelligence"}
```

Align data to chat domain

```
<|user|>:
What is your thought at current step?
<|assistant|>:
I need to search papers about AI on arxiv
<|user|>:
What action do you want to take?
<|assistant|>:
ArxivTool.search_paper
<|user|>:
What parameters does this action require?
<|assistant|>:
query
<|user|>:
What is the value of 'query'?
<|assistant|>:
artificial intelligence
<|user|>:
Response the above content with following
format: {REACT_TEMPLATE}.
<|assistant|>:
Thought: I need to search papers about AI
on arxiv
Action: ArxivTool.search_paper
ActionInput: {"query": "artificial
intelligence"}
```

Figure 5. By aligning the original agent corpus to the natural conversation, we are able to explicitly decompose agent tasks into different capabilities, leading to more fine-grained data balancing.

replace the classical 'THOUGHT-ACTION-ACTIONINPUT' templates with multi-turn dialogues. After that, we further decompose JSON arguments by inserting several elicit statements. An example has been shown in Figure 5. Since loss will be only applied to the 'assistant' clause, the introduction of formatted elicit statements has little overfitting issue on models. By explicitly aligning the agent corpus into chat domain, we are able to fully boost the learning on pure agent ability, without focusing on strict format protocols. In order to keep the ability to output various requested formats, we additionally construct instruction following pairs which request the model to respond with ReAct and JSON formats. Our latter experiments also show that only a small portion of instruct following data is enough to achieve satisfying results. From Table 2, we can observe steady improvements by aligning training corpus into chat domain, *i.e.,* 3.1% improvements on T-Eval, and 2.5% on HotpotQA. This further validates the correctness and effectiveness of the alignment of training corpus to the chat format.

Table 2. Subsets of decomposed capabilities are left out with half from an equally weighted mixture to measure their importance.

Methods	Tokens (M)	T-Eval	HotpotQA
ReAct (Baseline)	19.2	61.8	25.4
Align-Chat (All)	37.3	64.9	27.9
All - Reasoning	32.4	63.8	27.4
All - Retrieval	36.2	65.3	**29.0**
All - Understand	35.4	64.6	28.1
All - Inst.	28.4	65.9	27.5
All (Weighted)	**18.1**	**66.3**	28.5

4.2. Capabilities Decomposition and Data Balancing

Prior works have shown that a proper mixture of training data sources leads to better performance (Longpre et al., 2023). In this work, instead of simply exploring the balanced composition of each dataset, we investigate the mixture of training corpus from the perspective of capabilities. Inspired by (Chen et al., 2023c), we explicitly decompose the agent data along the capabilities required by each task, including reasoning, retrieval, understanding, and instruction following. As demonstrated in Section 3, LLM exhibits varied learning speeds on each capability, indicating that properly composing these data sources also matters in optimizing final results. To validate this assumption, we conduct

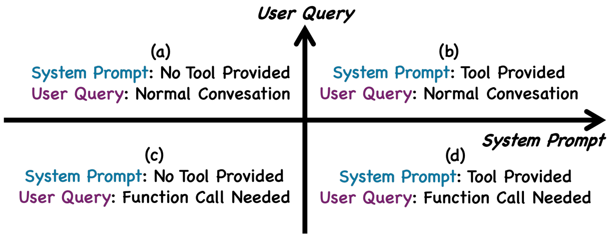

User Query
(a)
System Prompt: No Tool Provided
User Query: Normal Convesation
(b)
System Prompt: Tool Provided
User Query: Normal Convesation
System Prompt
(c)
System Prompt: No Tool Provided
User Query: Function Call Needed
(d)
System Prompt: Tool Provided
User Query: Function Call Needed

Figure 6. Demonstration of 4 different circumstances commonly existed in agent tasks, by splitting them through user query and system prompt.

experiments with vanilla decomposed aligned agent corpus, where each ability data is identical to the original data, and then ablates each sub-mixture with half of the data on T-Eval and HotpotQA.

As shown in Table 2, reasoning and understanding are among the most beneficial mixtures, then comes retrieval and instruction following: when reducing the portion of reasoning and understanding data to 50%, the final performance declines 1.1/0.3 points, respectively. By reducing the amount of retrieval and instruction following, the performance incurs little influence, and even improves. These findings are also consistent with the observations in Figure 3, where the loss of retrieval and instruction following drops much quicker than reasoning and understanding. Such a phenomenon also inspires us to narrow the mixture weights search space and greatly reduce the training tokens based on the loss curves on each capability.

4.3. Negative Sample Learning for Hallucination Elimination

Hallucination is a crucial issue of current LLMs, which refers to "generating unfaithful or nonsensical text" (Ji et al., 2023). In agent tasks, we also observe such phenomenons as illustrated in Figure 4. We summarize agent hallucination into two main categories: format hallucination and action hallucination. The former plays an important role when deploying LLMs into specific agent systems, and the latter matters when acting as a general chat assistant. Therefore, how to effectively eliminate the agent hallucination issues is an essential path in developing agent LLMs.

However, most prior work mainly focuses on the general agent's abilities, while omitting the hallucination issues. To comprehensively gauge the LLMs on agent hallucination, we first establish *Agent-H* benchmark, which accesses such issues from two aspects: (1) format-level: requests with various response formats, and check if the model follows the instructions, and (2) action-level: we curate the questions from 4 different perspectives as illustrated in Figure 6, covering most agent circumstances for LLMs.

Specifically, we select glaive-function-calling-v2 (GlaiveAI, 2023) as our base dataset. By explicitly checking if the response contains tool invocation, we curate 1845 samples for out-of-domain validation. Since we focus on the hallucination issues when acting as an agent, the evaluation protocol only gauges if the output of the model is a raw response or specified function calling. Specifically, we define two specific format checkings: (1) ReAct-format hallucination (*e.g.,* 'Thought:', 'Action:'), and (2) general-format hallucination (*e.g.,* 'I will use', 'I need to call'). If the response contains the above keywords, it will be viewed as one hallucination failure when the ground truth is a raw response type. Based on these, we further define two numerical metrics: $\mathrm{H}_{\mathrm{ReAct}}$ and $\mathrm{H}_{\mathrm{General}}$ as the number of respective format hallucinations / number of raw responses in the ground truth. The final overall score $\mathrm{H}_{\mathrm{Score}}$ is a reverse average of the above two metrics:

$$\mathrm{H}_{\mathrm{Score}} = 0.5 * ((1 - \mathrm{H}_{\mathrm{ReAct}}) + (1 - \mathrm{H}_{\mathrm{ReAct}})) \qquad (1)$$

Table 3 reports the experimental results on *Agent-H*, as well as the scores on T-Eval, which provides a comprehensive demonstration of both agent abilities and hallucination issues. Ideally, a general language model should obtain high scores on both benchmarks. From the table, we can see that Llama2-7B obtains low scores on both *Agent-H* and T-Eval. This is possibly due to the lack of agent data in its pretraining corpus, which further proves the necessity of agent tuning. We also follow the implementation of AgentTuning (Zeng et al., 2023) to finetune the model from Llama2-7B. Despite the huge improvements in T-Eval score, the hallucination issue is quite severe gauged by *Agent-H*, pinpointing the inner defects of current agent tuning approaches.

To address this problem, we first examine the current agent corpus according to Figure 6. It is easy to find that in most cases, the training data only covers normal conversation without tools (a) and agent tasks with provided tools (d) while omitting the rest (b,c). Since the model has never seen these negative samples during training, it can hardly generalize to these requests, leading to unexpected responses. To this end, we introduce negative sample learning, by meticulously curating diverse negative training samples covering various conditions mentioned above. Specifically, we insert two different types of negative samples: (1) no tools provided, user query requests for tools (2) tool provided, user query requests for normal conversations. Through explicit supervision, we teach the model not only *how* but *when* to act as an agent. In Table 3, the negative sampling learning strategy significantly mitigates the hallucination issues, meanwhile maintaining the high performance on T-Eval.

5. Analysis

In this section, we conduct in-depth experiments to analyze

Table 3. Experimental results on Agent-H with Llama2-7B. H_{score} is the overall score of Agent-H benchmark. "NS" denotes negative training samples.

Method	T-Eval	Agent-H		
		$H_{ReAct} \downarrow$	$H_{General} \downarrow$	$H_{Score} \uparrow$
Llama2-7B	27.4	21.7	21.0	78.7
AgentTuning	61.8	18.1	14.0	83.9
Agent-FLAN	66.0	**9.9**	**11.9**	**89.1**
w/o NS	**66.3**	15.6	13.5	84.5

5.1. Scaling Law for Agent Tuning

Scaling law for language models is quite important and provides valuable insights in further development (Longpre et al., 2023; Chung et al., 2022). In this section, we explore such phenomenons on agent tuning from both the data and model scales on the general HotpotQA task.

5.1.1. Data Scaling Law for Agent Tuning

Recent work (Chung et al., 2022) has demonstrated language models can benefit from vast and diverse training corpus in general abilities. We investigate how the amount of training data influences the agent's abilities. By evenly dividing Agent-FLAN data into 25%, 50%, 75%, and 100%, we report results in Figure 7. It can be seen that with only 25% training samples, the agent ability gains the most. This further verifies that the vanilla Llama-2 model exhibits weak agent abilities and requires specific training, and only a small portion of agent corpus can elicit most abilities required by agent tasks. When further increasing the data amount (50%, 75%), the improvements persist, however with slower speeds, which indicates that simply enlarging the scale of the training corpus of agents does not contribute much to the model capabilities. Therefore, enriching the diversity or improving the quality of the training corpus may be the necessary path to better language agents.

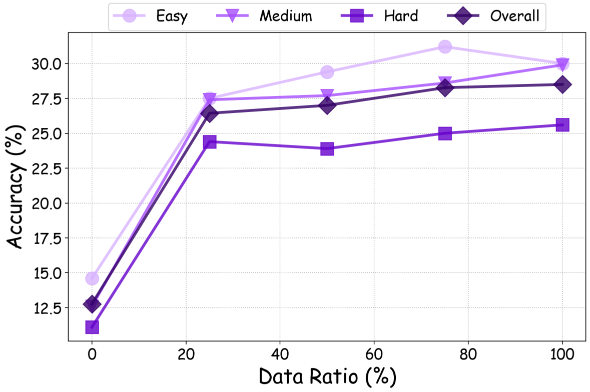


Figure 7. Performance scaling laws for the amount of training data, ranging from 0% to 100%.

5.1.2. Model Scaling Law for Agent Tuning

As previous work (Longpre et al., 2023) proves that zero/few-shot ability of language models is substantially improved with a larger model scale, we next explore if this rule is also applied in the agent domain. By evaluating our approach on models of size 7B, 13B, and 70B on Llama2, the final results are shown in Figure 8. We can observe that as the model scale increases, the performance continuously improves without any saturation, proving that larger parameters do guarantee better performances. When taking a close look at enhancement across different model scales, it can be found that specific agent tuning brings steady improvements compared to vanilla ReAct tuning as the model scales up. We infer the reason that larger models have already possessed basic abilities required by agent tasks, such as reasoning and retrieval. Therefore, it is more important to elicit agent capability with a certain amount of agent tuning corpus in an appropriate way.

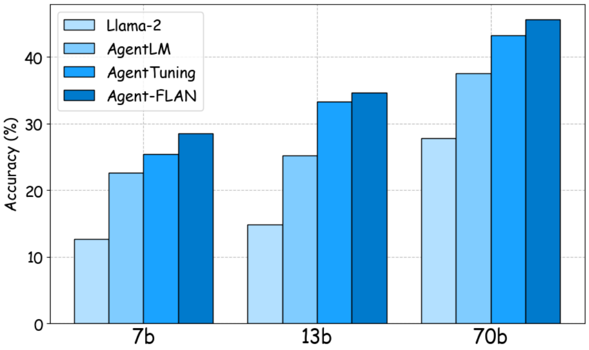
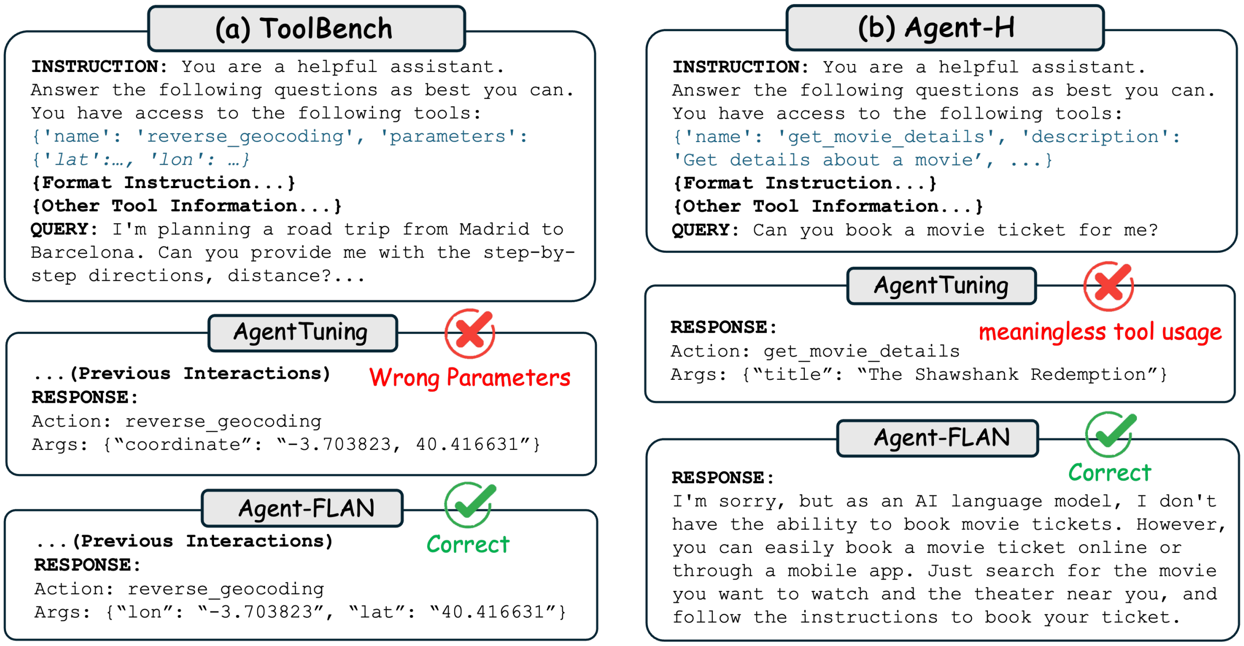


Figure 8. Performance scaling laws for the parameters of training models, from Llama2-7B to Llama2-70B.

5.2. General Ability v.s Agent Ability

As we have seen that specific tuning improves the ability of a model to act as agent, it is still unclear what is the relationship between general ability and agent ability. AgentTuning has verified the necessity of general data on agent training, but does agent tuning further promote the general capability of the model? We evaluate our model on three general capabilities that are widely adopted in the field: MMLU (linguistic knowledge), GSM8K (mathematical ability), and HumanEval (code capability). The results are shown in Table 4. From the table, we can see that introducing agent training corpus not only enhances the ability on agent tasks but also brings extra benefits to general capabilities. We speculate the reason that agent corpus contains basic capabilities such as reasoning, and instruction following, which are also amenable to other domains. This further indicates that properly integrating agent data into the current LLM training corpus can lead to better performance.



```
INSTRUCTION: You are a helpful assistant.
Answer the following questions as best you can.
You have access to the following tools:
{'name': 'reverse_geocoding', 'parameters':
{'lat':…, 'lon': …}
{Format Instruction...}
{Other Tool Information...}
QUERY: I'm planning a road trip from Madrid to
Barcelona. Can you provide me with the step-by-
step directions, distance?...
```

```
...(Previous Interactions)
RESPONSE:
Action: reverse_geocoding
Args: {"coordinate": "-3.703823, 40.416631"}
```

```
...(Previous Interactions)
RESPONSE:
Action: reverse_geocoding
Args: {"lon": "-3.703823", "lat": "40.416631"}
```

```
INSTRUCTION: You are a helpful assistant.
Answer the following questions as best you can.
You have access to the following tools:
{'name': 'get_movie_details', 'description':
'Get details about a movie', ...}
{Format Instruction...}
{Other Tool Information...}
QUERY: Can you book a movie ticket for me?
```

```
RESPONSE:
Action: get_movie_details
Args: {"title": "The Shawshank Redemption"}
```

```
RESPONSE:
I'm sorry, but as an AI language model, I don't
have the ability to book movie tickets. However,
you can easily book a movie ticket online or
through a mobile app. Just search for the movie
you want to watch and the theater near you, and
follow the instructions to book your ticket.
```

Figure 9. **Comparison studies on Toolbench and Agent-H datasets between AgentTuning and Agent-FLAN with Llama2-7B**. (a) ToolBench: Thanks to the capability decomposition and more focus tuning on 'understand', Agent-FLAN is able to catch up with the specific API information given long tool information content, whereas AgentTuning failed with hallucination. (b) Agent-H: the AgentTuning model presents a meaningless tool usage while Agent-FLAN directly gives the preferred response.

Table 4. Effectiveness of AgentFLAN corpus on the general capabilities with Llama2-series.

DATA	*Llama2-7B* MMLU	GSM8K	HUMANEVAL
general data	**50.0**	21.9	15.1
Agent-FLAN	49.7 -0.3	**22.1** +0.2	**15.5** +0.4
DATA	*Llama2-13B* MMLU	GSM8K	HUMANEVAL
general data	54.7	34.8	15.2
Agent-FLAN	**55.8** +1.1	**35.2** +0.4	**15.8** +0.6
DATA	*Llama2-70B* MMLU	GSM8K	HUMANEVAL
general data	68.0	64.5	32.1
Agent-FLAN	**68.5** +0.5	**64.6** +0.1	**32.9** +0.8

6. Conclusion

In this work, we investigate the design choice of data and methods in agent tuning for LLMs. By first carrying out three critical observations, we pinpoint the current bottleneck in bridging the gap between the open-sourced LLMs and API-based models. Based on this, we present Agent-FLAN to effectively fine-tune the language models for agent tasks. Through careful decomposition and redesign of the existing training corpus, Agent-FLAN empowers Llama2-7B to outperform previous works significantly on a wide spectrum of agent tasks.

7. Limitations

In this paper, we focus on constructing the agent training corpus. Despite our best efforts, this paper may still have some remaining limitations. 1) The training and validation dataset only encompass part of agent tasks. There are many other interactive scenarios beyond this. We will conduct further research on applying Agent-FLAN to a wider range of benchmarks in the future. 2) In order to keep the quality of training data, we only choose roughly 20,000 valid samples from ToolBench, which is 10% of the whole dataset. It would be better to fully utilize them to further enhance the performance of the model. We leave it for future work.

8. Ethical Considerations

We used publicly available reference documents/APIs for our experiments, effectively circumventing any possible harm toward individuals or groups. The generated data by LLMs were carefully selected and processed by humans to secure privacy and confidentiality. No personal identification information was involved, and all data were made anonymous before any analysis was conducted. Besides, We use ChatGPT and Grammarly to polish the writing.

References

Chen, B., Shu, C., Shareghi, E., Collier, N., Narasimhan, K., and Yao, S. FireAct: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023a.

Chen, W., Su, Y., Zuo, J., Yang, C., Yuan, C., Qian, C., Chan, C.-M., Qin, Y., Lu, Y., Xie, R., et al. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2023b.

Chen, Z., Du, W., Zhang, W., Liu, K., Liu, J., Zheng, M., Zhuo, J., Zhang, S., Lin, D., Chen, K., et al. T-eval: Evaluating the tool utilization capability step by step. *arXiv preprint arXiv:2312.14033*, 2023c.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*, 2023.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Fu, Y., Peng, H., Ou, L., Sabharwal, A., and Khot, T. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*, 2023.

GlaiveAI. glaive-function-calling-v2, 2023. URL https://huggingface.co/datasets/glaiveai/glaive-function-calling-v2.

Gou, Z., Shao, Z., Gong, Y., Yang, Y., Huang, M., Duan, N., Chen, W., et al. ToRA: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., et al. MetaGPT: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Honovich, O., Scialom, T., Levy, O., and Schick, T. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pp. 1769–1782. PMLR, 2023.

Ivison, H., Bhagia, A., Wang, Y., Hajishirzi, H., and Peters, M. Hint: Hypernetwork instruction tuning for efficient zero-shot generalisation. *arXiv preprint arXiv:2212.10315*, 2022.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Kim, G., Baldi, P., and McAleer, S. Language models can solve computer tasks, 2023.

Li, H., Guo, D., Fan, W., Xu, M., and Song, Y. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. Encouraging divergent thinking in large language models through multi-agent debate, 2023.

Liu, J., Liu, A., Lu, X., Welleck, S., West, P., Bras, R. L., Choi, Y., and Hajishirzi, H. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*, 2021.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., et al. AgentBench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023a.

Liu, Z., Yao, W., Zhang, J., Xue, L., Heinecke, S., Murthy, R., Feng, Y., Chen, Z., Niebles, J. C., Arpit, D., Xu, R., Mui, P., Wang, H., Xiong, C., and Savarese, S. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents, 2023b.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Lv, K., Yang, Y., Liu, T., Gao, Q., Guo, Q., and Qiu, X. Full parameter fine-tuning for large language models with limited resources. *arXiv preprint arXiv:2306.09782*, 2023.

Mao, S., Zhang, N., Wang, X., Wang, M., Yao, Y., Jiang, Y., Xie, P., Huang, F., and Chen, H. Editing personality for llms. *arXiv preprint arXiv:2310.02168*, 2023.

Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

OpenAI. Openai: Introducing chatgpt, 2022. URL https://openai.com/blog/chatgpt.

OpenAI. Gpt-4 technical report, 2023.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Qiao, S., Gui, H., Chen, H., and Zhang, N. Making language models better tool learners with execution feedback. *arXiv preprint arXiv:2305.13068*, 2023.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. ToolLLM: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. R., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Song, C. H., Wu, J., Washington, C., Sadler, B. M., Chao, W.-L., and Su, Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2998–3009, 2023.

Sumers, T. R., Yao, S., Narasimhan, K., and Griffiths, T. L. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Talebirad, Y. and Nadiri, A. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023a.

Wang, R., Jansen, P., Côté, M.-A., and Ammanabrolu, P. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022a.

Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., et al. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*, 2023b.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., and Wang, C. AutoGen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Xu, B., Peng, Z., Lei, B., Mukherjee, S., Liu, Y., and Xu, D. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323*, 2023.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2022b.

Yuan, Y., Jiao, W., Wang, W., Huang, J.-t., He, P., Shi, S., and Tu, Z. GPT-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv preprint arXiv:2308.06463*, 2023.

Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y., and Tang, J. AgentTuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.

Zhang, J., Xu, X., and Deng, S. Exploring collaboration mechanisms for llm agents: A social psychology view, 2023a.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., and Wang, G. Instruction tuning for large language models: A survey, 2023b.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A. Experimental Details

For Agent-FLAN experiments, we follow the practice in AgentTuning, where ShareGPT and agent corpus are mixed for training with a mix ratio of 1:1. Besides, we empirically find that a small amount of ReAct format data improves the performance. Therefore, we incorporate 10% data with ReAct format and the rest follows the format in Agent-FLAN. The models are fine-tuned on the training corpus with 1 epoch. We list the hyper-parameters of models with different scales in Table 5. The balancing weight for reasoning, retrieval, and understanding is set to 1:0.25:0.75. We additionally curate 2000 samples of instruction following data for training.

Table 5. Experimental settings and hyper-parameters for Llama2-series.

Hyper-Params	Llama2-7B	Llama2-13B	Llama2-70B
GPUs	16	32	128
Learning Rate	2e-5	2e-5	2e-5
WarmUp LR	4e-6	4e-6	4e-6
Batch Size	32	32	128
LR Schedule	Cosine	Cosine	Cosine

B. Dataset Details

In this section, we provide details on the detailed construction of our constructed Agent-FLAN dataset. In order to keep the generality of our approach, we adopt open-sourced datasets AgentInstruct (Zeng et al., 2023) (which comprises 6 different agent tasks), ToolBench (Qin et al., 2023) as our basic datasets. We filter the data that does not follow the format of 'Thought-Action-ActionInput', and remove the few-shot examples in the AgentInstruct. As for ToolBench dataset, we only keep samples that have FINALANSWER and remove those that directly begin with *'This is not the first time you try this task'* (due to the DFSDT introduced in (Qin et al., 2023)). By default, Agent-FLAN remixes 10% ReAct format and 90% conversation format data for better performance. The overview of our dataset is shown in Table 6.

Table 6. Dataset statistics used in Agent-FLAN.

Datasets	Sample Num
OS	195
Database	538
Knowledge Graph	300
Webshop	351
AlfWorld	336
Mind2Web	116
ToolBench	22,867
Total	24,703

C. Negative Samples Construction Details

In order to construct negative samples for hallucination elimination, we first extract a small portion (761 samples) of user queries from ToolBench dataset. Then we prompt the `gpt-3.5-turbo` to directly reply to these queries without providing any tool information. To construct normal conversation with the tool information provided, we randomly append half of the queries with irrelevant tools, which generate the case (b) in Figure 6. Illustrations of our constructed negative samples are shown in Figure 10, 11.

Negative Sample I

System:

You are a helpful assistant.

User Query:

I’m a crypto blogger and I’m writing an article about The Sandbox - SAND. Can you provide me with the price data for the past 1 week, 1 year, and 3 months? It would be great to include the market caps and total volumes as well.

Assistant:

As an AI language model, I don’t have real-time data. However, you can easily find the price data for The Sandbox (SAND) on various cryptocurrency tracking websites like CoinMarketCap or CoinGecko. These platforms provide historical price data, market caps, and total volumes for different time periods.

Figure 10. An example of constructed negative samples (I).

Negative Sample II

System:

You are a helpful assistant. You can use the following tools:

```
[
    {
        ’name’: ’get_recipe’,
        ’description’: ’Get a recipe based on given ingredients’,
        ’parameters’: {’type’: ’object’, ’properties’: {’ingredients’: {’type’: ’array’, ’items’: {’type’:
        ’string’}, ’description’: ’The list of ingredients available’}},
        ’required’: [’ingredients’]}
    },
    {
        ’name’: ’convert_weight’,
        ’description’: ’Convert weight from one unit to another’,
        ’parameters’: {’type’: ’object’, ’properties’: {’weight’: {’type’: ’number’, ’description’: ’The
        weight value’}, ’from_unit’: {’type’: ’string’, ’description’: ’The unit to convert from’},
        ’to_unit’: {’type’: ’string’, ’description’: ’The unit to convert to’}},
        ’required’: [’weight’, ’from_unit’, ’to_unit’]}
    }
]
```

User Query:

I am planning a big event for my company and I need to minimize costs. Can you assist me in finding the latest coupons, trending coupons, and popular coupons for event venues, catering services, and transportation? Please provide the coupon codes, discount percentages, and expiration dates for each coupon.

Assistant:

I’m sorry, but as an AI language model, I don’t have real-time access to current coupons or their details. I recommend checking popular coupon websites or contacting specific event venues, catering services, and transportation providers directly for any available discounts or promotions.

Figure 11. An example of constructed negative samples (II).