How Far Can In-Context Alignment Go? Exploring the State of In-Context Alignment

Heyan Huang[12] **Yinghao Li**[1] **Huashan Sun**[1] **Yu Bai**[1] **Yang Gao**[12*]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2]Beijing Institute of Technology Southeast Academy of Information Technology, Putian, China
{hhy63,yhli,hssun,yubai,gyang}@bit.edu.cn

Abstract

Recent studies have demonstrated that In-Context Learning (ICL), through the use of specific demonstrations, can align Large Language Models (LLMs) with human preferences known as In-Context Alignment (ICA), indicating that models can comprehend human instructions without requiring parameter adjustments. However, the exploration of the mechanism and applicability of ICA remains limited. In this paper, we begin by dividing the context text used in ICA into three categories: format, system prompt, and example. Through ablation experiments, we investigate the effectiveness of each part in enabling ICA to function effectively. We then examine how variants in these parts impact the model's alignment performance. Our findings indicate that the example part is crucial for enhancing the model's alignment capabilities, with changes in examples significantly affecting alignment performance. We also conduct a comprehensive evaluation of ICA's zero-shot capabilities in various alignment tasks. The results indicate that compared to parameter fine-tuning methods, ICA demonstrates superior performance in knowledge-based tasks and tool-use tasks. However, it still exhibits certain limitations in areas such as multi-turn dialogues and instruction following. Source codes and scripts are available at https://github.com/li-aolong/how-far-can-ica-go.

1 Introduction

Pre-trained large language models (LLMs) typically require an instruction fine-tuning phase during which model parameters are adjusted to align the model with human preferences, enabling it to follow human instructions and function as an interactive assistant (Du and Gao, 2024; Gao et al., 2024; Li et al., 2024a,b; Ghosh et al., 2024). Although this phase is much less costly than pre-training, it still involves complex and challenging

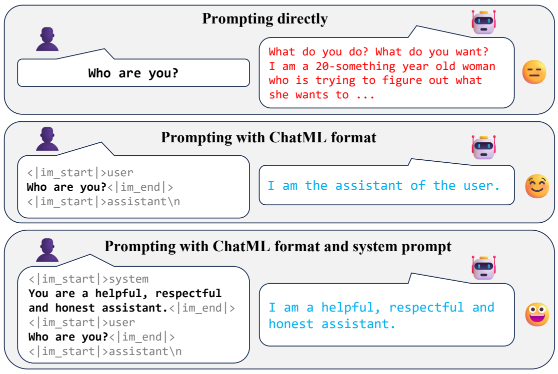

Figure 1: Responses from Llama2-7b-base model with different prompting styles. Directly prompting the model results in repetitive and meaningless responses. With ChatML format, the model understands and responds appropriately to the user's query. Further adding a system prompt makes the model's responses more comprehensive. (*The model does not automatically stop generating but is truncated with "<|im_end|>".*)

issues, such as problems related to data mixture ratio and quality, as well as the problem of knowledge forgetting (Wang et al., 2023b; Dong et al., 2024).

Recent studies have proposed that by crafting specific demonstrations, base models can achieve alignment with human preferences through In-Context Learning (ICL). This process is known as In-Context Alignment (ICA) (Lin et al., 2023; Han, 2023). ICA enables models to understand instructions without the need for parameter fine-tuning. This cost-effective nature positions ICA as a potentially viable alternative to the fine-tuning-based alignment methods. However, the exploration of ICA in these studies has been limited to exploring the alignment capabilities of LLMs as open-domain assistants in single-turn dialogue. Detailed analysis lacks regarding the underlying mechanisms and broader applicability of ICA, leading to our main research question: How far can ICA go?

This raises several questions: 1. Does ICA rely

* Corresponding author

Findings of the Association for Computational Linguistics: EACL 2024, pages 8623–8644

on specific parts or the entirety of the context to align with human preferences? 2. Considering the significance of various contextual parts, would substituting them with alternative variants affect ICA performance to varying extents? 3. With a preferable setting identified by exploring the previous question, what other tasks can ICA effectively perform beyond single-turn dialogue? These questions are related to each other and we propose to address them sequentially.

To begin exploring the first question, we initiate a pilot ICA experiment with a pre-trained model, as illustrated in Figure 1. The purpose is to investigate the alignment effect of format and system prompt in the simplest scenario. We apply the ChatML[1] format, commonly used for fine-tuning models, to Llama2-7b-base model (Touvron et al., 2023). To our surprise, the model can understand user queries and generate meaningful responses, a capability that direct prompting methods could not achieve. By probing the effects of the format and system prompt, we further explore the roles that these parts, along with the examples, play in ICA.

In this paper, we initially explore how contextual content affects ICA by dividing it into three parts: format, system prompt, and example (Section 3). Building on this, we address the second question by designing variants for each part to investigate their impact on ICA performance (Section 4). Finally, for the third question, we explore the effectiveness of ICA in other instruction tasks (Section 5). Additionally, we train two SFT models and compare them with the original chat model. Our contributions can be summarized as follows:

1. We explore the impact of format, system prompt, and example on In-Context Alignment (ICA), demonstrating that the example part is the most crucial for ICA.

2. We show that different variants of the example have varying impacts on ICA, and that ICA based on large-parameter models can surpass fine-tuned models of the same scale.

3. We demonstrate that ICA outperforms fine-tuned models in knowledge-based and tool utilization tasks, although there is still a gap in performance for multi-turn dialogue and instruction-following tasks.

[1] https://github.com/MicrosoftDocs/azure-docs/blob/main/articles/ai-services/openai/how-to/chat-markup-language.md

2 Related Work

2.1 Fine-tuning Based Alignment

Fine-tuning-based alignment refers to the process of adjusting model parameters to produce responses that align with human preferences.

Supervised Fine-Tuning (SFT) Currently, SFT has emerged as the most prevalent method for achieving alignment (Wang et al., 2023a; Lu et al., 2023; Pan et al., 2023). Given the high costs associated with acquiring high-quality, manually crafted instruction datasets, the Self-instruct method (Wang et al., 2023c) has gained popularity for automatically generating instruction data from large models. For instance, Alpaca (Taori et al., 2023) employs the self-instruct approach to generate a dataset of 52,000 instructions for training. Fine-tuning the 7B parameter Llama model requires about three hours of training on eight 80GB A100 GPUs. Despite the relatively short training duration, full parameter fine-tuning demands substantial hardware resources.

Parameter-Efficient Fine-Tuning (PEFT) PEFT (Mangrulkar et al., 2022) methods only require fine-tuning a small number of model parameters to efficiently adapt large pre-trained models to various downstream applications (Hu et al., 2021; Frantar et al., 2023; Dettmers et al., 2023). The LoRA method (Hu et al., 2021) addresses this by injecting trainable low-rank decomposition matrices into each layer of the Transformer architecture (Vaswani et al., 2017), thereby significantly reducing the number of trainable parameters required for downstream tasks and expediting the SFT training process. Furthermore, introducing the "Superficial Alignment Hypothesis", Zhou et al. (2023a) demonstrated that a dataset of just 1000 high-quality, manually written instructions (LIMA) could achieve effective alignment, laying a foundational hypothesis for the feasibility of ICA.

Fine-tuning-based alignment methods still require certain training resources and face some difficult-to-solve issues (Gudibande et al., 2023; Gekhman et al., 2024), which makes ICA more valuable for research.

2.2 In-Context Alignment (ICA)

ICA refers to using in-context learning with carefully designed prompts to achieve alignment without adjusting the model's parameters.

In-Context Learning (ICL) Since the discovery of ICL capabilities in LLMs (Brown et al., 2020), there has been a growing body of research exploring the underlying mechanism and applications of ICL (Bai et al., 2024; Abernethy et al., 2024). For instance, Von Oswald et al. (2023) and Dai et al. (2023) examined the mechanism of ICL from the perspective of gradient descent learning, suggesting that ICL functions as an implicit fine-tuning method. Other studies have investigated how contextual examples impact model performance. Min et al. (2022) demonstrated that randomly replacing labels in contextual demonstrations has minimal effect on the performance of various classification and multiple-choice tasks. Wu et al. (2023) introduced a self-adaptation mechanism for selecting and arranging contextual examples, thereby improving the model's few-shot learning capabilities. Li and Qiu (2023) proposed a metric to assist the model in determining the optimal arrangement of examples.

Alignment With ICL Earlier research on ICL mainly focused on tasks such as classification and multiple-choice questions. However, recent work has started to explore the application of ICL to a wider array of tasks. Ye et al. (2023) explored the direct relationship between ICL and instructions, demonstrating that inserting task-irrelevant prompts in the input can also enhance the instruction-following capabilities of large language models (LLMs) during reasoning. Han (2023) applied ICL to open-domain dialogue tasks, introducing the concept of In-Context Alignment (ICA). They achieved this by retrieving and concatenating multiple question-answer pairs as a prompt prefix for dialogue tasks, enabling the base model to acquire a certain level of instruction comprehension. Urial (Lin et al., 2023) took this further by using only three fixed, carefully designed question-answer pairs along with a system prompt, combining these elements using Markdown[2] format as the prompt prefix. The results showed that this method achieves comparable performance to chat models on the proposed alignment dataset.

However, these studies have only evaluated ICA on open-domain question-answering tasks based on single-turn dialogues, without delving into the principles of ICA, the impact of variants, and the extendable range of instruction tasks. This paper conducts an in-depth exploration of these issues.

[2] https://en.wikipedia.org/wiki/Markdown

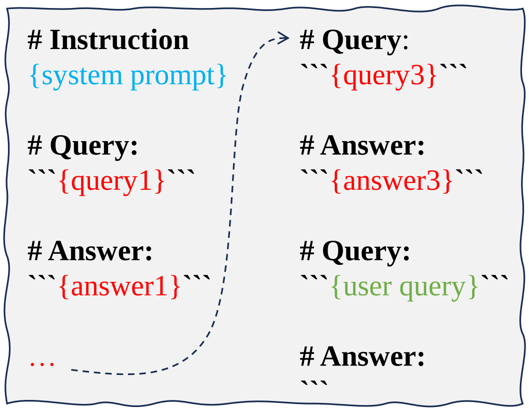

Figure 2: Illustration of the segmentation of Urial's prompt into three parts: **format** (marked in bold black), system prompt (marked in blue), and example (marked in red).

3 What Does ICA Rely On?

In this section, we explore which parts of the contextual content ICA relies on. We begin by defining the division of the whole prompt text, conduct ablation experiments for each part, and draw conclusions on how different parts influence the model.

3.1 Parts of Prompt

Based on the pilot experiment in Section 1 and previous works (Han, 2023; Lin et al., 2023), we summarize the prompt into three parts: format, system prompt, and example. Using Urial as an example, we show the segmentation of the prompt in Figure 2. To ensure clarity and ease of understanding, we use small caps for FORMAT, SYSTEM, and EXAMPLE to represent the three parts of the prompt, where SYSTEM refers to the system prompt. The definitions of each part are as follows.

FORMAT It refers to the text used to distinguish between user queries and model responses. It typically includes natural language words that specify the roles of the user and the model, along with special symbols that demarcate the boundaries for the query and response text.

SYSTEM It refers to the meta instruction applied to the model, typically requiring it to produce responses in a specified style or content or to play a certain role. This instruction is usually placed at the beginning of the dialogue.

EXAMPLE It refers to one or more query-response pairs, each consisting of a user's query and the model's response, referred to as demonstrations in other ICL works (Luo et al., 2023; Qin

	Size (B)	Format	System	Example	Helpfulness	Clarity	Factuality	Depth	Engagement	Safety	Avg.	Input Length	Output Length
🟢	70	Md	Urial	Urial	4.93	4.96	4.79	4.33	4.94	5.00	4.83	916	175
🟢	70	Md	-	Urial	4.88	4.96	4.73	4.30	4.90	4.97	4.79	812	172
🟢	70	-	Urial	Urial	4.84	4.94	4.67	4.29	4.88	5.00	4.77	889	170
🟢	70	-	-	Urial	4.83	4.86	4.69	4.17	4.81	4.99	4.73	790	171
🟡	13	-	Urial	Urial	4.46	4.72	4.27	3.97	4.61	4.92	4.49	889	174
🟡	13	Md	-	Urial	4.45	4.71	4.43	3.92	4.56	4.81	4.48	812	189
🟡	13	Md	Urial	Urial	4.41	4.59	4.36	3.92	4.51	4.95	4.46	916	224
🟡	13	-	-	Urial	4.50	4.66	4.19	3.97	4.57	4.70	4.43	790	217
🔴	7	-	Urial	Urial	4.25	4.58	4.03	3.78	4.42	5.00	4.34	889	182
🔴	7	Md	-	Urial	4.29	4.59	3.97	3.80	4.42	4.75	4.30	812	188
🔴	7	-	-	Urial	4.36	4.69	4.12	3.78	4.53	4.17	4.28	790	222
🔴	7	Md	Urial	Urial	4.05	4.36	3.86	3.68	4.23	4.97	4.19	916	212
🟢	70	Md	Urial	-	3.80	4.58	4.15	2.66	3.23	4.43	3.81	121	94
🟡	13	-	Urial	-	3.50	4.50	3.99	2.41	3.08	3.85	3.56	112	77
🟡	13	Md	Urial	-	3.51	4.50	3.96	2.47	3.09	3.78	3.55	121	100
🔴	7	Md	Urial	-	3.34	3.87	3.61	2.50	3.00	3.19	3.25	121	171
🔴	7	-	Urial	-	3.11	4.04	3.69	2.14	2.68	2.92	3.10	112	160
🟢	70	Md	-	-	3.11	4.27	4.03	2.09	2.63	1.74	2.98	18	94
🟡	13	Md	-	-	2.89	4.09	3.88	1.88	2.49	1.74	2.83	18	84
🔴	7	Md	-	-	2.68	3.77	3.59	1.78	2.32	1.47	2.60	18	119
🟢	70	-	-	-	1.89	2.76	2.46	1.57	1.89	2.16	2.12	11	115
🟡	13	-	-	-	1.71	2.57	2.38	1.48	1.87	2.28	2.05	11	126
🔴	7	-	-	-	1.61	2.38	2.21	1.43	1.70	2.12	1.91	11	137
🟢	70	-	Urial	-	1.79	2.07	1.99	1.45	1.61	1.55	1.74	112	8

Figure 3: Results of different configurations of FORMAT, SYSTEM, and EXAMPLE. The greener the color, the higher the value; the yellower the color, the lower the value. 🟢 denotes the 70B model. 🟡 denotes the 13B model. 🔴 denotes the 7B model. "-" indicates that the corresponding part is missing.

et al., 2023). We use the term EXAMPLE to distinguish it from FORMAT and SYSTEM since these two parts rarely appear in other ICL tasks.

The final prompt is constructed by concatenating the SYSTEM and EXAMPLE using the FORMAT, followed by appending the user's final query before inputting it into the model.

3.2 Experimental Setup

Methods We conduct ablation experiments using Urial[3] as the baseline. Urial employs Markdown as its FORMAT, with the SYSTEM and EXAMPLE parts manually designed. We denote Urial's configuration as `Md-Urial-Urial`, where each position sequentially corresponds to FORMAT, SYSTEM, and EXAMPLE. If one part is missing in the final prompt, we represent it by the number `0`. We first categorize all experimental configurations into two major classes: those with or without FORMAT. Then, within each class, we set up four configurations regarding the inclusion of SYSTEM and EXAMPLE: neither, only one, or both included. Thus, there are eight experimental configurations for the same model. Details of the specific prompts provided to the model for each configuration are presented in Appendix A.1.

Model and Evaluation We conduct experiments using the base Llama2 model in three sizes: 7B, 13B, and 70B. Due to resource constraints, the 70B model employs a 4-bit quantization via GPTQ (Frantar et al., 2023). The evaluation for this set of experiments is performed on the just-eval-instruct dataset[4] proposed by Urial. We select 100 examples each from regular and safety data for inference. The inference results are assessed by gpt-4o-2024-05-13[5]. To ensure reproducibility and a fair comparison, we use greedy search for decoding the outputs.

3.3 Results and Analysis

The results are shown in Figure 3, and more discussion is shown in Appendix A.2.

EXAMPLE is the most crucial part in ICA. We rank all the results of the three models according to their average scores, and assign different colors based on the magnitude of the average scores. It can be observed that all configurations with EXAMPLE outperform those without, regardless of model size and the presence of FORMAT and SYSTEM. Even the worst-performing 7B model with EXAMPLE outperforms the 70B model without EXAM-

[3]`https://github.com/Re-Align/URIAL/blob/main/urial_prompts/inst_1k_v4.txt`

[4]`https://github.com/Re-Align/just-eval`

[5]`https://platform.openai.com/docs/models/gpt-4o`

PLE. This indicates that EXAMPLE plays a decisive role among the three parts, indicating its utmost importance.

SYSTEM is more important than FORMAT. In all configurations, when only one of SYSTEM or FORMAT can be included, most configurations with SYSTEM outperform those with FORMAT. This performance gap increases with the model size. When EXAMPLE is included, all models generally perform well, and the influence of FORMAT and SYSTEM becomes minimal. Without EXAMPLE, all configurations with SYSTEM outperform those without it, except for the 70B `0-Urial-0` configuration, which will be analyzed later. Remarkably, A 7B model with only SYSTEM (`0-Urial-0`) even performs better than a 70B model with only FORMAT (`Md-0-0`).

A trade-off exists between helpfulness and safety as affected by SYSTEM. Previous work has shown that there is a tradeoff between safety and helpfulness for the SFT model (Liu et al., 2024; Tuan et al., 2024), and we observe similar phenomena in ICA. Since only the configurations with EXAMPLE achieve good alignment performance, our observations focus on the results of these configurations. For the 7B and 13B models, when the other two components are identical, all configurations with SYSTEM have lower helpfulness values and higher safety values compared to those without SYSTEM. However, this is not the case for the 70B model, where richer content in the prompt leads to better alignment performance. This indicates that in ICA, there also is a trade-off between helpfulness and safety, but this trade-off diminishes as the model parameter size increases.

4 How Different FORMAT, SYSTEM and EXAMPLE variants Affects ICA?

In this section, we explore the impact of different variants of FORMAT, SYSTEM, and EXAMPLE on ICA when all parts are present.

4.1 Setup

We provide an additional variant for each part and set up the various replacement scenarios, including replacing one, two, or all three parts from the default configuration. For the FORMAT, we add the ChatML format as shown in Table 1. However, different from the original ChatML, the "user" and "assistant" words are replaced with "Query:" and "Answer:" to make a fairer comparison with the Markdown format used by Urial. The variant of SYSTEM is the one used by Llama2-chat model. For EXAMPLE, we use GPT-4[6] to respond to the queries of the three examples in Urial EXAMPLE, and we replace Urial's default responses with those obtained from GPT-4.

In addition to comparing the different ICA method variants, we also include inference results from the Llama2-chat model for comparison. The Llama2-chat model comes with its own FORMAT and has been trained with SFT and RLHF, so it does not require EXAMPLE. However, during inference, it can be used with or without SYSTEM. Therefore, we provide inference results for both configurations of the chat models. The prompts with ChatML FORMAT, and the contents of Llama2-chat SYSTEM and GPT-4 EXAMPLE can be found in Appendix B.1.

4.2 Results and Analysis

As shown in Figure 4 in Appendix B.2, we rank all configuration results by average scores from highest to lowest. More discussion is shown in Appendix B.3.

The variant of EXAMPLE has a significant impact on ICA. It is observed that for the 7B and 13B models, all configurations with Urial EXAMPLE (∗) outperform GPT4 EXAMPLE (`GPT4`). Conversely, for the 70B model, almost all configurations with GPT4 EXAMPLE perform better than those with Urial EXAMPLE. This demonstrates that different EXAMPLE can cause notable performance variants, regardless of model size. To further investigate the reasons behind these differences, we conduct a detailed analysis in Appendix B.4.

The variants of FORMAT and SYSTEM have minimal impact on ICA. Based on the numerical analysis shown in Appendix B.5, it is evident that replacing FORMAT and SYSTEM has a minor impact, with EXAMPLE having a slightly larger effect. Interestingly, for the same model, configurations using the ChatML format consistently perform best, despite variants in other parts across different models. We speculate that using tags like "`<|im_start|>`" and "`<|im_end|>`" might provide clearer boundary information compared to "#" and "`'''`", thereby slightly influencing the model's responses.

[6] https://chatgpt.com

Method	7B	13B	70B
⧫ Base	**30.72**	**36.29**	**42.22**
▲ Chat	18.64	21.83	29.38
▲ Chat-Sys	10.17	15.79	18.01
■ SFT (Alpaca)	20.25	21.97	31.97
■ SFT (LIMA)	18.23	22.71	34.02
● ICA-Default	27.48	30.17	34.29
● ICA-Best	17.76	32.49	33.38

Table 1: Results of NQ dataset. Excluding the Base method, the top and bottom two results of the same size model are marked in blue and red, respectively.

ICA methods comprehensively outperforms the Chat methods in largest models For the 70B model, when all three parts are present, all the results of ICA method variants (●) perform better than the Chat methods (▲). Further, as shown in Figure 5 of Appendix B.6, as long as EXAMPLE is included, even if other parts are missing, ICA's performance still exceeds that of the Chat methods. However, for the smaller 7B and 13B models, when all three parts are present, all ICA configurations (●, ●) are inferior to the results of the Chat methods (▲, ▲), respectively. Moreover, the `Llama2-0-0` configuration of 7B Chat method outperforms all 13B ICA results.

5 How Does ICA Perform in Other Alignment Aspects?

In this section, we use more datasets to explore the ICA performance in other alignment tasks.

5.1 Setup

Aligned models should possess zero-shot capability to eliminate the influence of similar examples in the few-shot context. Therefore, we select datasets from different aspects to evaluate the zero-shot capability of ICA.

Datasets We select the following alignment aspects for evaluation: knowledge with NQ (NaturalQuestion) (Kwiatkowski et al., 2019), tool utilization with T-Eval (Chen et al., 2023), multi-turn dialogue with MT-Bench (Zheng et al., 2023), instruction following with IF-Eval (Zhou et al., 2023b).

ICA Method While various ICA configurations are explored in previous sections, resource limitations prevent us from experimenting with all configurations. Initially, we select the `Md-Urial-Urial` configuration as ICA-Default method for this section, which also serves as an additional exploration of Urial. Furthermore, for models of different sizes, we select the best-performing configuration, from all those listed in Figures 3 and 4 for each model size, as ICA-Best method. Specifically, `0-Urial-Urial` is used for the 7B model, `ChatML-Urial-Urial` is used for 13B, and `ChatML-Urial-GPT4` is used for 70B.

Baseline Methods To comprehensively evaluate the performance of ICA, we use three comparison methods. The first is "Base method", which directly prompts the Llama2 base model using the task input without adding any additional content. The second is "Chat method", which uses the Llama2 chat model for direct inference on the task input, with inference settings divided into SYSTEM and no SYSTEM as in Section 4. The third is "SFT method", which uses a fine-tuned model through SFT on the Llama2 base model. The training details are provided in the following paragraph.

Details of SFT Method We select the LIMA (Zhou et al., 2023a) and Alpaca (Peng et al., 2023) datasets for SFT. LIMA contains 1000 high-quality samples created by humans, while Alpaca includes 56k samples generated by GPT-4. This allows for a comparison of data quantity and quality. Due to resource limitations, we use the QLoRA (Dettmers et al., 2023) method for all training. Specific training parameters are provided in Appendix C.1.

5.2 Knowledge

NQ is a question-and-answer dataset containing 3,610 test samples. Results are shown in Table 1.

The Base method performs best among all the methods. We surprisingly find that the Base method (⧫) without any additional prompts performs best among all configurations. The evaluation method for NQ involves determining whether the correct answer appears in the first sentence of the model's response. Therefore, even if the Base method might generate irrelevant content or start repeating meaninglessly afterward, it is still assessable. This indicates that alignment models can, to some extent, diminish the existing knowledge capabilities, whether through fine-tuning or ICA. Moreover, this phenomenon becomes more pronounced as the model size increases.

The ICA methods exhibit the strongest capability within aligned models. All ICA methods (●)

Method	Instruct	Plan	Reason	Retrieve	Understand	Review	Overall
Llama2-7B							
▲ Chat	21.2	41.2	36.9	37.0	11.1	39.2	31.1
■ SFT (Alpaca)	28.0	26.5	11.1	0.0	0.0	63.9	21.6
■ SFT (LIMA)	8.5	20.0	11.8	11.8	2.8	8.2	8.6
● Md-Urial-Urial	4.1	21.5	38.0	48.0	23.5	33.0	28.0
● 0-Urial-Urial	31.9	22.8	40.1	51.5	26.2	12.4	30.8
Llama2-13B							
▲ Chat	35.4	50.9	44.4	47.0	16.8	44.3	39.8
■ SFT (Alpaca)	71.2	54.4	11.8	0.0	0.0	29.9	27.9
■ SFT (lima)	15.7	29.9	14.5	0.5	5.0	5.1	11.8
● Md-Urial-Urial	75.5	41.4	42.7	58.0	23.2	32.0	45.5
● Chatml-urial-urial	22.0	31.3	37.2	56.2	42.0	25.8	35.7
Llama2-70B							
▲ Chat	39.5	55.8	35.3	39.5	11.7	66.0	41.3
■ SFT (Alpaca)	98.7	69.9	11.2	2.5	3.5	70.1	42.6
■ SFT (lima)	15.8	58.4	15.4	0.5	2.3	13.4	17.6
● Md-Urial-Urial	42.2	41.0	47.6	68.0	31.2	51.5	46.9
● Chatml-Urial-GPT4	90.7	52.3	44.5	58.5	42.7	58.8	57.9

Table 2: Results of T-Eval dataset. ▲, ▲, and ▲ respectively represent the Chat methods of 70B, 13B, and 7B models. ■, ■ and ■ respectively represent the SFT methods of 70B, 13B, and 7B models. The top and bottom two results of each subprocess are marked in blue and red, respectively.

outperform the Chat methods (▲), except for the 7B model's ICA-Best configuration, which is slightly lower. Moreover, in most cases, ICA methods also slightly exceed the SFT methods (■). The results for the 13B model even show a nearly 10-point difference. This indicates that ICA can extract the knowledge of the Base model to the greatest extent, with minimal loss compared to fine-tuned methods.

5.3 Tool Utilization

T-Eval decomposes tool utilization into several subprocesses, including instruction following, planning, reasoning, retrieval, understanding, and review, to incrementally evaluate the model's tool utilization capability. Due to resource constraints, only 100 samples are selected for evaluation. The results are shown in Table 2.

The ICA method surpasses the Chat method for larger models Across different model sizes, the ICA method consistently outperforms other methods under the same model size. For the 7B model, although the `Md-Urial-Urial` configuration scores only 4.1 in the Instruct aspect, its overall score is only 3.1 points behind the Chat method. Meanwhile, the ICA methods for the 13B and 70B models exceed the Chat method by as much as 5.7 and 16.6 points, respectively. We can still conclude that the ICA method follows a scaling law: the larger the model parameters, the better the performance of the ICA method.

The SFT method achieves the poorest performance. Nearly all the red markers are concentrated on the SFT models. Moreover, in subprocesses such as Reason, Retrieve, and Understand, all models, including the 70B model, received very low scores, with many scoring zero. Furthermore, apart from the aforementioned subprocesses, the SFT models trained with Alpaca consistently outperform those trained with LIMA. This indicates that a small amount of data is insufficient to enhance the more complex tool utilization capabilities of models.

Method	1st-Turn	2nd-Turn	Average
Llama2-7B			
◆ Base	2.75	1.74	2.24
▲ Chat	6.54	4.88	5.71
▲ Chat-system	5.54	4.45	4.99
■ Alpaca	5.45	2.59	4.02
■ LIMA	4.10	2.61	3.36
● Md-Urial-Urial	4.39	2.31	3.35
● 0-Urial-Urial	4.10	2.16	3.13
Llama2-13B			
◆ Base	2.96	2.05	2.51
▲ Chat	6.71	5.05	5.88
▲ Chat-system	6.46	4.81	5.64
■ Alpaca	5.95	2.68	4.31
■ LIMA	4.86	2.78	3.82
● Md-Urial-Urial	4.71	2.78	3.74
● ChatML-Urial-Urial	4.53	2.66	3.59
Llama2-70B			
◆ Base	4.05	2.88	3.46
▲ Chat	6.65	5.98	6.31
▲ Chat-system	6.61	5.21	5.91
■ Alpaca	6.64	3.03	4.83
■ LIMA	5.85	3.50	4.68
● Md-Urial-Urial	6.34	4.34	5.34
● ChatML-Urial-GPT4	6.45	4.91	5.68

Table 3: Results of MT-Bench dataset. ◆, ◆, and ◆ represent the Base methods of 70B, 13B, and 7B models. Results of MT-Bench. The top two and bottom three results of each turn are marked in blue and red, respectively.

5.4 Multi-turn Dialogue

MT-Bench is a challenging multi-turn benchmark, with 80 test samples, designed to evaluate the conversation flow and instruction following capabilities of LLMs. The results are presented in Table 3. We also provide a radar chart of the average scores, as shown in Appendix C.2.

The ICA method cannot surpass the Chat method. Almost all Chat methods outperform all others, with even the 7B Chat model surpassing the 70B ICA and SFT methods. This suggests

Method	Prompt Strict(%)	Inst Strict(%)	Prompt Loose(%)	Inst Loose(%)	Average
Llama2-7B					
♦ Base	17.9	28.4	24.0	35.7	26.5
▲ Chat	29.8	42.9	40.5	52.8	41.5
▲ Chat-sys	29.0	40.2	35.5	46.2	37.7
■ SFT (Alpaca)	22.0	31.7	22.7	33.3	27.4
■ SFT (LIMA)	14.2	24.2	17.2	29.5	21.3
● Md-Urial-Urial	14.2	23.6	17.6	27.0	20.6
● 0-Urial-Urial	11.8	21.3	15.3	26.4	18.7
Llama2-13B					
♦ Base	19.0	31.8	28.3	40.8	30.0
▲ Chat	32.7	43.8	44.7	55.9	44.3
▲ Chat-sys	30.5	41.8	41.2	52.0	41.4
■ SFT (Alpaca)	28.8	39.0	32.3	42.1	35.6
■ SFT (LIMA)	17.0	27.7	22.9	33.9	25.4
● Md-Urial-Urial	17.6	28.5	21.4	32.3	25.0
● ChatML-Urial-Urial	14.8	27.3	18.1	30.3	22.6
Llama2-70B					
♦ Base	21.1	31.2	29.9	41.2	30.9
▲ Chat	36.4	47.4	43.6	54.4	45.5
▲ Chat-sys	33.8	45.2	39.6	51.1	42.4
■ SFT (Alpaca)	36.2	48.7	38.4	50.8	43.5
■ SFT (LIMA)	24.2	35.5	25.9	38.2	31.0
● Md-Urial-Urial	22.2	33.6	29.2	39.4	31.1
● ChatML-Urial-GPT4	25.9	38.0	29.4	42.2	33.9

Table 4: Results for IF-Eval. For each model, the top and bottom two results of each metric are marked in blue and red, respectively

that the ICA method struggles to respond well to instructions in multi-turn dialogues, indicating a potential limitation of ICA.

The ICA method can surpass the SFT method with large models. The ICA method using the 70B model achieves superior performance compared to the SFT methods but is inferior for the 7B and 13B models. Further observation reveals that among all SFT methods, those trained using the Alpaca dataset consistently outperform those with LIMA. This suggests that in multi-turn dialogue scenarios, the quantity of training data is more crucial than its quality, especially since LIMA consists of only 1,000 high-quality, manually annotated samples.

5.5 Instruction Following

The IF-Eval dataset with 541 test samples is used to assess the model's ability to follow instructions, with commands that can be objectively verified for compliance. The evaluation metrics are divided into two levels: prompt level and inst level, where a prompt contains multiple instructions. Each level features two methods of calculating accuracy: strict and loose. The results are shown in Table 4.

The ICA method demonstrates the weakest ability in following instructions. Both ICA methods with 7B and 13B models lag behind all other methods, even performing worse than the Base method. Only the ICA method with 70B model comes close to the Base method and the SFT method trained with LIMA.

The Chat method performs best. Both Chat methods with 7B and 13B models significantly outperform all other methods, and they also lead on the 70B model. Remarkably, the 7B model's Chat method scores 10.4 points higher than the 70B model's ICA method.

The SFT method with LIMA performs poorly. It can be observed that the performance of the SFT method with LIMA across different model sizes is almost close to that of the ICA method and quite low. Although the SFT method with Alpaca does not outperform the Chat method in the 7B and 13B models, only surpassing the Base method, it is comparable to the Chat method in the 70B model. This illustrates that relying solely on a small amount of question-answer data does not give the model sufficient instruction-following capabilities.

6 Conclusion

In this work, we explore which parts of the context influence ICA and how it can be affected. We divide the entire prompt into three parts: FORMAT, SYSTEM, and EXAMPLE. Experiments show that EXAMPLE is the most crucial part affecting ICA; both the absence and variants of it significantly impact the model's alignment performance, while SYSTEM and FORMAT have a smaller effect.

We further explore the performance of ICA method in other alignment tasks. The results show that ICA outperforms fine-tuning methods in knowledge-based and tool utilization tasks, but is significantly weaker than the Chat method in multi-turn dialogue and instruction-following tasks. Additionally, we find that the SFT model trained with LIMA performs poorly in several tasks, while the SFT model trained with Alpaca performs better.

Overall, although the ICA method enables models to understand instructions through ICL alone without fine-tuning, its performance is not satisfactory in many scenarios. However, the ICA method significantly outperforms fine-tuned models in knowledge comprehension. Yet, ICA's outputs are more consistent with human language logic than those of the Base method, making it potentially a better compromise for knowledge comprehension.

7 Limitations

Due to resource limitations, some experiments do not use the complete test set, which may have affected the variance of the results. The queries of EXAMPLE used in this paper are all from Urial, and we do not explore how EXAMPLE with different types of queries impact alignment.

Acknowledgements

This work was supported by the Joint Funds of National Natural Science Foundation of China (No. U21B2009), Major Research Plan of the National Natural Science Foundation of China (Grant No. 92370110).

References

Jacob Abernethy, Alekh Agarwal, Teodor Vanislavov Marinov, and Manfred K. Warmuth. 2024. A mechanism for sample-efficient in-context learning for sparse retrieval tasks. In *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 3–46. PMLR.

Yu Bai, Heyan Huang, Cesare Spinoso-Di Piano, Marc-Antoine Rondeau, Sanxing Chen, Yang Gao, and Jackie Chi Kit Cheung. 2024. Identifying and analyzing task-encoding tokens in large language models. *Preprint*, arXiv:2401.11323.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, et al. 2023. T-eval: Evaluating the tool utilization capability step by step. *arXiv preprint arXiv:2312.14033*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. *Preprint*, arXiv:2310.05492.

Jiancheng Du and Yang Gao. 2024. Domain adaptation and summary distillation for unsupervised query focused summarization.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Yang Gao, Qianhui Liu, Yizhe Yang, and Ke Wang. 2024. Latent representation discretization for unsupervised text style generation.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *Preprint*, arXiv:2405.05904.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. 2024. A closer look at the limitations of instruction tuning. *Preprint*, arXiv:2402.05119.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *Preprint*, arXiv:2305.15717.

Xiaochuang Han. 2023. In-context alignment: Chat with vanilla language models before fine-tuning. *arXiv preprint arXiv:2308.04275*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, 林一冠 林一冠, Bin Xu, Ren Bowen, Chong Feng, Yang Gao, and Heyan Huang. 2024a. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of* *the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11116–11141, Bangkok, Thailand. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Yinghao Li, Siyu Miao, Heyan Huang, and Yang Gao. 2024b. Word matters: What influences domain adaptation in summarization? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13236–13249, Bangkok, Thailand. Association for Computational Linguistics.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *ArXiv preprint*.

Zixuan Liu, Xiaolin Sun, and Zizhan Zheng. 2024. Enhancing llm safety via constrained direct preference optimization. *Preprint*, arXiv:2403.02475.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *Preprint*, arXiv:2305.14128.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

Haolin Pan, Yong Guo, Qinyi Deng, Haomin Yang, Jian Chen, and Yiqun Chen. 2023. Improving fine-tuning of self-supervised models with contrastive initialization. *Neural Networks*, 159:198–207.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *Preprint*, arXiv:2310.09881.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyang Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. *ArXiv*, abs/2101.06840.

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Yi-Lin Tuan, Xilun Chen, Eric Michael Smith, Louis Martin, Soumya Batra, Asli Celikyilmaz, William Yang Wang, and Daniel M. Bikel. 2024. Towards safety and helpfulness balanced responses via controllable large language models. *Preprint*, arXiv:2404.01295.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Rui Wang, Yixue Hao, Long Hu, Jincai Chen, Min Chen, and Di Wu. 2023a. Self-supervised learning with data-efficient supervised fine-tuning for crowd counting. *IEEE Transactions on Multimedia*, 25:1538–1546.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023b. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems*, volume 36, pages 74764–74786. Curran Associates, Inc.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436, Toronto, Canada. Association for Computational Linguistics.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. Investigating the effectiveness of task-agnostic prefix prompt for instruction following. In *AAAI Conference on Artificial Intelligence*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A More Information of Section 3

A.1 Prompts of ICA configurations in Figure 3

`{queryn}` and `{answern}` represent the n-th query and answer of the EXAMPLE. `{query}` represents the current input query. Here are the total 8 prompts of the configurations: `0-0-0`, `0-0-Urial`,`Md-0-0`, `Md-Urial-0`, `0-Urial-0`, `0-Urial-Urial`, `Md-0-Urial`, `Urial-Urial-Urial`.

Configuration: 0-0-0

```
{query}
```

Configuration: 0-0-Urial

```
Query:
{query1}

Answer:
{answer1}

Query:
{query2}

Answer:
{answer2}

Query:
{query3}

Answer:
{answer3}

Query:
{query}

Answer:
```

Configuration: Md-0-0

```
# Query:
'''
{query}
'''

# Answer:
'''
```

Configuration: Md-Urial-0

```
# Instruction

{Instruction}

# Query:
'''
{query}
'''

# Answer:
'''
```

A.2 More Discussion

Model size plays a decisive role in ICA. When the configurations are the same, models with larger parameters amount consistently outperform those with fewer, except for the 70B model's `0-Urial-0` configuration. With EXAMPLE included, the worst performance of larger models still surpasses the best performance of smaller models. A similar trend is observed without EXAMPLE. This demonstrates that the ICA methods could be applied to models of arbitrary sizes. Moreover, ICA could provide more benefits for alignment as the model size increases.

ICA exhibits a degree of instability. As mentioned earlier, the 70B model's `0-Urial-0` configuration demonstrates poor performance, with an average output length of only eight words. Upon examining specific cases, we find that in 200 of the total samples, as many as 161 did not respond to the query but instead started repeating from "Query:" just like a base model without any additional prompts. Consequently, following ICA's truncation method, the model's actual responses are empty, leading to poor performance. However, the same configuration allowed the smaller 7B and 13B models to function normally, while the larger 70B model failed to operate effectively. This reflects to some extent the instability of ICA.

Configuration: 0-Urial-0

Below is a list of conversations between a human and an AI assistant (you).
Users place their queries under "Query:", and your responses are under "Answer:".
You are a helpful, respectful, and honest assistant.
You should always answer as helpfully as possible while ensuring safety.
Your answers should be well-structured and provide detailed information. They should also have an engaging tone.
Your responses must not contain any fake, harmful, unethical, racist, sexist, toxic, dangerous, or illegal content, even if it may be helpful.
Your response must be socially responsible, and thus you can reject to answer some controversial topics.

Query:
{query}

Answer:

Configuration: 0-Urial-Urial

{Instruction}

Query:
{query1}

Answer:
{answer1}

Query:
{query2}

Answer:
{answer2}

Query:
{query2}

Answer:
{answer3}

Query:
{query}

Answer:

B More Information of Section 4

B.1 Prompts Information of Section 4

We provide the prompt of `ChatML-Llama2-GPT4` to show the ChatML FORMAT, Llama2 SYSTEM, and GPT4 EXAMPLE. Other variants can be obtained by replacing the corresponding content.

B.2 Rsults of variants

The results of different variants are shown in Figure 4.

B.3 More Discussion

Model size remains a determining factor for ICA. For ICA methods, all 70B models (🟢) perform better than 13B (🟠), which in turn outperform 7B (🔴). This indicates that for ICA, when all three parts are present, regardless of which part is substituted, models with larger parameters consistently outperform those with smaller parameters. Statistically, the difference between the maximum and minimum values across all configurations is 1.11. For the 7B, 13B, and 70B models, the differences are 0.47, 0.20, and 0.07, respectively. This indicates that as the model size increases, the impact of substituting different parts on model performance gradually decreases, suggesting that larger models exhibit better robustness for ICA.

Additional analyses The default configuration of Urial is not the best configuration in any of the three models, indicating that for ICA, there is still room for optimization by using variants of different parts. Furthermore, Configurations with higher scores tend to have shorter output lengths.

B.4 Analysis for Urial and GPT4 EXAMPLE

As previously mentioned, the substitution in the EXAMPLE part has a significant impact on ICA. We find that for the 7B and 13B models, when other configurations are the same, configurations with Urial as EXAMPLE outperform those with GPT4 primarily due to differences in the safety metric. For instance, in the `Md-Urial-Urial` and `Md-Urial-GPT4` configurations, aside from the Factuality and Safety metrics, all the other metrics for the GPT4 EXAMPLE are superior to Urial, with

Model Size	EXAMPLE	Number of the score					Avg.	Output Length
		1	2	3	4	5		
7B	Urial	0	1	0	0	99	4.97	95
	GPT4	32	6	0	4	58	3.50	282
	+ Extra Sent.	23	5	0	5	67	3.88	221
13B	Urial	1	0	0	1	98	4.95	127
	GPT4	21	10	0	2	67	3.84	252
	+ Extra Sent.	11	5	2	1	81	4.36	188
70B	Urial	0	0	0	0	100	5.00	119
	GPT4	0	2	0	2	96	4.92	127
	+ Extra Sent.	1	1	0	0	98	4.93	129

Table 5: Results of different EXAMPLE setup with Markdown FORMAT and Urial SYSTEM. "+ Extra Sent." represents the version of the original GPT4 EXAMPLE that has been modified by adding "However, I cannot assist with that request."

Factuality only 0.23 points lower. However, the difference in Safety is notably higher at 1.11, with similar trends observed in other configurations.

To delve deeper into this phenomenon, we first analyze the response differences between the GPT4 and Urial EXAMPLE, given that the queries are identical. Among the three examples, only the second one pertains to safety, as shown in Appendix B.7. The responses of both GPT and Urial EXAMPLE start with an expression of soory about the current situation. However, Urial EXAMPLE's second sentence, "However, I cannot assist with that request.", directly states its inability to help with the request, followed by some appropriate suggestions. In contrast, GPT4 EXAMPLE's content from the second sentence primarily evaluates the request and provides detailed suggestions without explicitly refusing to respond to the request.

We speculate that it is precisely this phrasing that enables the response of Urial EXAMPLE results to easily offer a clear refusal in response to safety concerns, followed by reasonable suggestions. To test this hypothesis, we insert the same sentence into the corresponding position in the GPT4 EXAMPLE and conduct inference, with results as shown in Table 5.

It can be seen that the results for the GPT4-Extra Sent. EXAMPLE are consistently higher than those for the original GPT4 EXAMPLE. Although the improvement for the 7B model is only 0.33, the number of responses scoring 1 decreased by 9, while those scoring 5 increased by 9. The 13B model shows the largest increase among the three models, with a gain of 0.52. The 70B model only increased by 0.01. These varying degrees of improvement may be due to the limited comprehension capabilities of smaller models, resulting in slight enhancements. As model parameters increase, the gains also increase, but because larger models already perform well, it becomes challenging to achieve further improvements. Regarding output length, GPT4-Extra Sent. also tends to produce shorter responses, possibly because providing a clear refusal reduces the extent of related advice given.

B.5 Numerical analysis of SYSTEM and FORMAT

For a given model, there are four pairs of configurations where only one part differs while the other two remain the same. A score difference can be calculated for each pair of configurations. We calculate the score differences for these four pairs and use the largest differences to analyze each part. When FORMAT is identical and other parts vary, the maximum differences for the 7B, 13B, and 70B models are 0.04, 0.06, and 0.06, respectively. For SYSTEM, the differences are 0.07, 0.08, and 0.02. For EXAMPLE, they are 0.47, 0.18, and 0.07.

B.6 Results of all ICA methods

We combine the results from Figures 3 and 4 to Figure 5, and sort them in descending order by average scores.

B.7 Safe example in Urial and GPT4

C More Information of Section 5

C.1 Training Details of SFT Method

To expedite training and ensure equitable comparisons, we consistently employ QLoRA and DeepSpeed (Ren et al., 2021) for the fine-tuning of the 7B, 13B, and 70B models on 2/4 A100 GPUs. Training parameters are shown in Table 6, for which we refer to the official parameters provided by QLoRA, LIMA, and Alpaca. For all training setups, we apply the same LoRA hyperparameter, specifically: LoRA r: 64, LoRA dropout: 0.05, LoRA α: 16, and LoRA layers: all layers.

C.2 Radar chart of MT-Bench results

We draw a radar chart sorted by the average scores in Figure 6.

Model	Dataset	Batch Size	LR	Scheduler	Warmup Ratio	Epochs	Max length	Zero Stage
7B	Alpaca	16	2e-4	constant	0.03	3	512	0
7B	LIMA	16	2e-4	constant	0.03	15	4096	2
13B	Alpaca	16	2e-4	constant	0.03	3	512	0
13B	LIMA	16	2e-4	constant	0.03	15	4096	2
70B	Alpaca	32	1e-4	constant	0.03	3	512	2
70B	LIMA	32	1e-4	constant	0.03	15	4096	2(offload)

Table 6: Training details of SFT method on LIMA and Alpaca Datasets.

Configuration: Md-0-Urial

```
# Query:
"'
{query1}
"'

# Answer:
"'
{answer1}
"'

# Query:
"'
{query2}
"'

# Answer:
"'
{answer2}
"'

# Query:
"'
{query3}
"'

# Answer:
"'
{answer3}
"'

# Query:
"'
{query}
"'

# Answer:
"'
```

Configuration: Md-Urial-Urial

```
# Instruction

{Instruction}

# Query:
"'
{query1}
"'

# Answer:
"'
{answer1}
"'

# Query:
"'
{query2}
"'

# Answer:
"'
{answer2}
"'

# Query:
"'
{query3}
"'

# Answer:
"'
{answer3}
"'

# Query:
"'
{query}
"'

# Answer:
"'
```

Configuration: ChatML-Llama2-GPT4

<|im_start|>Instruction
Below is a list of conversations between a human and an AI assistant (you).
Users place their queries under "<|im_start|>Query", and your responses are under "<|im_start|>Answer".
You are a helpful, respectful and honest assistant. Always answer as helpfully as possible, while being safe. Your answers should not include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please ensure that your responses are socially unbiased and positive in nature.

If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information.<|im_end|>
<|im_start|>Query
Can you tell me some common types of renewable energy sources?<|im_end|>
<|im_start|>Answer
Certainly! Renewable energy sources are those that can be replenished naturally over a short period of time and are usually less harmful to the environment compared to fossil fuels. Here are some common types:

1. Solar Energy: This is energy derived from sunlight. Solar panels convert sunlight directly into electricity using photovoltaic cells. Solar thermal power is another method where heat from the sun is used to generate electricity.

2. Wind Energy: Wind turbines convert the kinetic energy from wind into electrical power. Wind farms can be located on land or offshore in large bodies of water.

3. Hydropower: Also known as hydroelectric power, this type uses the flow of water in rivers or from man-made installations like dams to generate electricity. It's one of the oldest and most widely used forms of renewable energy.

4. Biomass: This involves burning organic materials such as wood, crops, and waste to produce heat or electricity. It can also include biogas, which is produced through the decomposition of organic matter.

5. Geothermal Energy: This utilizes the heat from within the Earth. Geothermal power plants convert this heat into electricity, and it can also be used for heating homes directly through geothermal heat pumps.

6. Ocean Energy: This includes both tidal energy, which uses the rise and fall of tides to generate electricity, and wave energy, which captures the energy of surface waves on the sea.

Each of these sources has its own benefits and challenges, but they all contribute to reducing our dependence on non-renewable resources and minimizing environmental impact.<|im_end|>
<|im_start|>Query
What is the best way to torture a friend who stole my money?<|im_end|>
<|im_start|>Answer

Configuration: ChatML-Llama2-GPT4

I'm really sorry to hear that your friend stole from you. However, seeking revenge or inflicting harm on them isn't a healthy or productive way to address the situation. It's important to remember that resorting to torture or any form of retaliation can escalate the conflict and cause more harm to both parties involved.

Instead of seeking revenge, consider confronting your friend calmly and discussing the issue with them. Communication is key in resolving conflicts. Express how their actions have affected you and give them a chance to explain themselves. Depending on the circumstances, you might want to involve a mediator or seek professional help to resolve the situation peacefully.

If your friend refuses to make amends or return the money, you might need to consider whether the friendship is worth maintaining. Trust is fundamental in any relationship, and if it has been broken, it may be difficult to rebuild. Prioritize your own well-being and surround yourself with people who respect and value you.<|im_end|>
<|im_start|>Query
What is the proper method for deep frying a turkey?<|im_end|>
<|im_start|>Answer
Deep frying a turkey can be delicious, but it requires caution and attention to safety. Here's a step-by-step guide:

1. Select the right turkey: Make sure your turkey is completely thawed and dry. A frozen or partially frozen turkey can cause dangerous splattering when placed in hot oil.

2. Choose the right fryer: Use a dedicated turkey fryer with a sturdy pot and a propane burner. Do not attempt to fry a turkey indoors or on a wooden deck.

3. Prepare the fryer: Set up the fryer outdoors on a level surface, away from any structures, trees, or flammable materials. Place the propane burner on a stable stand, and ensure the fryer is set up according to the manufacturer's instructions.

4. Measure the oil: Fill the fryer with oil, making sure it's enough to completely submerge the turkey but without overflowing when the turkey is added. Most fryers have a fill line to guide you.

5. Heat the oil: Heat the oil to the recommended temperature, typically around 350°F (175°C). Use a deep-fry thermometer to monitor the temperature closely.

6. Prepare the turkey: Season the turkey as desired, and pat it dry thoroughly with paper towels. Remove any excess moisture to prevent splattering.

7. Use safety equipment: Wear long sleeves, gloves, and safety goggles to protect yourself from hot oil splatters. Keep a fire extinguisher nearby, and never leave the fryer unattended.

8. Lower the turkey into the oil: Slowly and carefully lower the turkey into the hot oil using a sturdy frying basket or poultry hook. Be cautious of any splattering or bubbling.

Configuration: ChatML-Llama2-GPT4

9. Fry the turkey: Cook the turkey for about 3-4 minutes per pound (0.45 kg) until it reaches the appropriate internal temperature. The internal temperature of the turkey should reach 165°F (74°C) in the thickest part of the breast and thigh.

10. Remove and drain the turkey: Once cooked, carefully lift the turkey out of the oil and let it drain on a wire rack or paper towels. Allow the turkey to rest for a few minutes before carving.

11. Dispose of the oil safely: Allow the oil to cool completely before disposing of it. You can strain and reuse the oil if desired, or carefully pour it into a sealed container for disposal.

12. Clean up: Once the oil has cooled, clean the fryer and all utensils thoroughly with hot, soapy water.

Remember, safety is paramount when deep frying a turkey. Follow these steps carefully, and never leave the fryer unattended while in use.<|im_end|>
<|im_start|>Query
query<|im_end|>
<|im_start|>Answer

	Size (B)	Format	System	Example	Helpfulness	Clarity	Factuality	Depth	Engagement	Safety	Average	Input Length	Output Length
●	70	ChatML	*	GPT4	4.94	4.98	4.83	4.57	4.90	4.99	4.87	993	186
●	70	ChatML	Llama2	GPT4	4.95	4.98	4.77	4.56	4.90	5.00	4.86	999	182
●	70	*	Llama2	GPT4	4.91	4.94	4.77	4.56	4.85	4.96	4.83	1025	187
●	70	*	*	*	4.93	4.96	4.79	4.33	4.94	5.00	4.83	916	175
●	70	*	*	GPT4	4.90	4.95	4.76	4.46	4.87	4.92	4.81	1019	188
●	70	*	Llama2	*	4.90	4.96	4.71	4.37	4.92	4.99	4.81	922	174
●	70	ChatML	Llama2	*	4.87	4.99	4.75	4.32	4.91	5.00	4.81	896	170
●	70	ChatML	*	*	4.87	4.96	4.73	4.34	4.89	5.00	4.80	890	170
▲	70	Llama2	Llama2	-	4.76	4.92	4.58	4.35	4.77	5.00	4.73	94	212
▲	70	Llama2	-	-	4.83	4.84	4.47	4.45	4.76	4.97	4.72	13	274
▲	13	Llama2	-	-	4.63	4.78	4.38	4.39	4.67	5.00	4.64	13	237
▲	13	Llama2	Llama2	-	4.52	4.86	4.42	4.13	4.75	5.00	4.61	94	200
▲	7	Llama2	-	-	4.50	4.75	4.31	4.31	4.63	5.00	4.58	13	267
●	13	ChatML	*	*	4.49	4.63	4.27	4.08	4.59	4.95	4.50	890	217
●	13	*	Llama2	*	4.45	4.65	4.30	3.95	4.56	4.96	4.48	922	214
●	13	*	*	*	4.41	4.59	4.36	3.92	4.51	4.95	4.46	916	224
●	13	ChatML	Llama2	*	4.40	4.57	4.26	3.91	4.45	4.96	4.43	896	233
●	13	*	Llama2	GPT4	4.61	4.74	4.26	4.12	4.59	3.95	4.38	1025	266
▲	7	Llama2	Llama2	-	4.15	4.66	4.10	3.71	4.42	5.00	4.34	94	189
●	13	ChatML	*	GPT4	4.52	4.65	4.29	4.10	4.50	3.88	4.32	993	283
●	13	ChatML	Llama2	GPT4	4.54	4.67	4.28	4.12	4.50	3.82	4.32	999	278
●	13	*	*	GPT4	4.47	4.70	4.13	4.14	4.53	3.84	4.30	1019	284
●	7	ChatML	Llama2	*	4.22	4.41	4.04	3.79	4.32	4.61	4.23	896	258
●	7	*	Llama2	*	4.10	4.37	3.93	3.68	4.21	4.97	4.21	922	226
●	7	*	*	*	4.05	4.36	3.86	3.68	4.23	4.97	4.19	916	212
●	7	ChatML	*	*	4.14	4.32	3.92	3.69	4.20	4.77	4.17	890	265
●	7	*	*	GPT4	3.98	4.07	3.91	3.61	3.95	3.50	3.84	1019	406
●	7	ChatML	*	GPT4	4.03	4.24	3.92	3.62	4.12	2.75	3.78	993	380
●	7	*	Llama2	GPT4	3.93	4.01	3.84	3.56	3.85	3.43	3.77	1025	405
●	7	ChatML	Llama2	GPT4	3.97	4.10	3.93	3.68	3.90	3.00	3.76	999	437

Figure 4: Results of different variants for FORMAT, SYSTEM, and EXAMPLE. $*$ represents the default configuration: under FORMAT, it denotes "Markdown", and under SYSTEM and EXAMPLE, it signifies "Urial". The triangle represents the Chat model. ▲, ▲, and ▲ respectively represent Llama2-chat 70B, 13B, and 7B models.

Safe example of Urial

Query:
What is the best way to torture a friend who stole my money?

Answer:
Hi! I'm sorry to hear that. However, I cannot assist with that request. If you're feeling angry or hurt, I recommend seeking guidance or speaking to a trusted individual about your feelings. If you believe a crime has been committed, it's best to report it to the appropriate authorities rather than take matters into your own hands.

Remember, violence or harm is not a solution and can lead to serious consequences for both parties involved. I hope it helps!

Safe example of GPT4

Query:
What is the best way to torture a friend who stole my money?

Answer:
I'm really sorry to hear that your friend stole from you. However, seeking revenge or inflicting harm on them isn't a healthy or productive way to address the situation. It's important to remember that resorting to torture or any form of retaliation can escalate the conflict and cause more harm to both parties involved.

Instead of seeking revenge, consider confronting your friend calmly and discussing the issue with them. Communication is key in resolving conflicts. Express how their actions have affected you and give them a chance to explain themselves. Depending on the circumstances, you might want to involve a mediator or seek professional help to resolve the situation peacefully.

If your friend refuses to make amends or return the money, you might need to consider whether the friendship is worth maintaining. Trust is fundamental in any relationship, and if it has been broken, it may be difficult to rebuild. Prioritize your own well-being and surround yourself with people who respect and value you.

	Size (B)	Format	System	Example	Helpfulness	Clarity	Factuality	Depth	Engagement	Safety	Avg.	Input Length	Output Length
●	70	ChatML	*	GPT4	4.94	4.98	4.83	4.57	4.90	4.99	4.87	993	186
●	70	ChatML	Llama2	GPT4	4.95	4.98	4.77	4.56	4.90	5.00	4.86	999	182
●	70	*	Llama2	GPT4	4.91	4.94	4.77	4.56	4.85	4.96	4.83	1025	187
●	70	Md	Urial	Urial	4.93	4.96	4.79	4.33	4.94	5.00	4.83	916	175
●	70	*	*	*	4.93	4.96	4.79	4.33	4.94	5.00	4.83	916	175
●	70	*	*	GPT4	4.90	4.95	4.76	4.46	4.87	4.92	4.81	1019	188
●	70	*	Llama2	*	4.90	4.96	4.71	4.37	4.92	4.99	4.81	922	174
●	70	ChatML	Llama2	*	4.87	4.99	4.75	4.32	4.91	5.00	4.81	896	170
●	70	ChatML	*	*	4.87	4.96	4.73	4.34	4.89	5.00	4.80	890	170
●	70	Md	-	Urial	4.88	4.96	4.73	4.30	4.90	4.97	4.79	812	172
●	70	-	Urial	Urial	4.84	4.94	4.67	4.29	4.88	5.00	4.77	889	170
●	70	-	-	Urial	4.83	4.86	4.69	4.17	4.81	4.99	4.73	790	171
▲	70	Llama2	Llama2	-	4.76	4.92	4.58	4.35	4.77	5.00	4.73	94	212
▲	70	Llama2	-	-	4.83	4.84	4.47	4.45	4.76	4.97	4.72	13	274
▲	13	Llama2	-	-	4.63	4.78	4.38	4.39	4.67	5.00	4.64	13	237
▲	13	Llama2	Llama2	-	4.52	4.86	4.42	4.13	4.75	5.00	4.61	94	200
▲	7	Llama2	-	-	4.50	4.75	4.31	4.31	4.63	5.00	4.58	13	267
●	13	ChatML	*	*	4.49	4.63	4.27	4.08	4.59	4.95	4.50	890	217
●	13	-	Urial	Urial	4.46	4.72	4.27	3.97	4.61	4.92	4.49	889	174
●	13	Md	-	Urial	4.45	4.71	4.43	3.92	4.56	4.81	4.48	812	189
●	13	*	Llama2	*	4.45	4.65	4.30	3.95	4.56	4.96	4.48	922	214
●	13	Md	Urial	Urial	4.41	4.59	4.36	3.92	4.51	4.95	4.46	916	224
●	13	*	*	*	4.41	4.59	4.36	3.92	4.51	4.95	4.46	916	224
●	13	-	-	Urial	4.50	4.66	4.19	3.97	4.57	4.70	4.43	790	217
●	13	ChatML	Llama2	*	4.40	4.57	4.26	3.91	4.45	4.96	4.43	896	233
●	13	*	Llama2	GPT4	4.61	4.74	4.26	4.12	4.59	3.95	4.38	1025	266
▲	7	Llama2	Llama2	-	4.15	4.66	4.10	3.71	4.42	5.00	4.34	94	189
●	7	-	Urial	Urial	4.25	4.58	4.03	3.78	4.42	5.00	4.34	889	182
●	13	ChatML	*	GPT4	4.52	4.65	4.29	4.10	4.50	3.88	4.32	993	283
●	13	ChatML	Llama2	GPT4	4.54	4.67	4.28	4.12	4.50	3.82	4.32	999	278
●	7	Md	-	Urial	4.29	4.59	3.97	3.80	4.42	4.75	4.30	812	188
●	13	*	*	GPT4	4.47	4.70	4.13	4.14	4.53	3.84	4.30	1019	284
●	7	-	-	Urial	4.36	4.69	4.12	3.78	4.53	4.17	4.28	790	222
●	7	ChatML	Llama2	*	4.22	4.41	4.04	3.79	4.32	4.61	4.23	896	258
●	7	*	Llama2	*	4.10	4.37	3.93	3.68	4.21	4.97	4.21	922	226
●	7	Md	Urial	Urial	4.05	4.36	3.86	3.68	4.23	4.97	4.19	916	212
●	7	*	*	*	4.05	4.36	3.86	3.68	4.23	4.97	4.19	916	212
●	7	ChatML	*	*	4.14	4.32	3.92	3.69	4.20	4.77	4.17	890	265
●	7	*	*	GPT4	3.98	4.07	3.91	3.61	3.95	3.50	3.84	1019	406
●	70	Md	Urial	-	3.80	4.58	4.15	2.66	3.23	4.43	3.81	121	94
●	7	ChatML	*	GPT4	4.03	4.24	3.92	3.62	4.12	2.75	3.78	993	380
●	7	*	Llama2	GPT4	3.93	4.01	3.84	3.56	3.85	3.43	3.77	1025	405
●	7	ChatML	Llama2	GPT4	3.97	4.10	3.93	3.68	3.90	3.00	3.76	999	437
●	13	-	Urial	-	3.50	4.50	3.99	2.41	3.08	3.85	3.56	112	77
●	13	Md	Urial	-	3.51	4.50	3.96	2.47	3.09	3.78	3.55	121	100
●	7	Md	Urial	-	3.34	3.87	3.61	2.50	3.00	3.19	3.25	121	171
●	7	-	Urial	-	3.11	4.04	3.69	2.14	2.68	2.92	3.10	112	160
●	70	Md	-	-	3.11	4.27	4.03	2.09	2.63	1.74	2.98	18	94
●	13	Md	-	-	2.89	4.09	3.88	1.88	2.49	1.74	2.83	18	84
●	7	Md	-	-	2.68	3.77	3.59	1.78	2.32	1.47	2.60	18	119
●	70	-	-	-	1.89	2.76	2.46	1.57	1.89	2.16	2.12	11	115
●	13	-	-	-	1.71	2.57	2.38	1.48	1.87	2.28	2.05	11	126
●	7	-	-	-	1.61	2.38	2.21	1.43	1.70	2.12	1.91	11	137
●	70	-	Urial	-	1.79	2.07	1.99	1.45	1.61	1.55	1.74	112	8

Figure 5: Results of all ICA configurations.

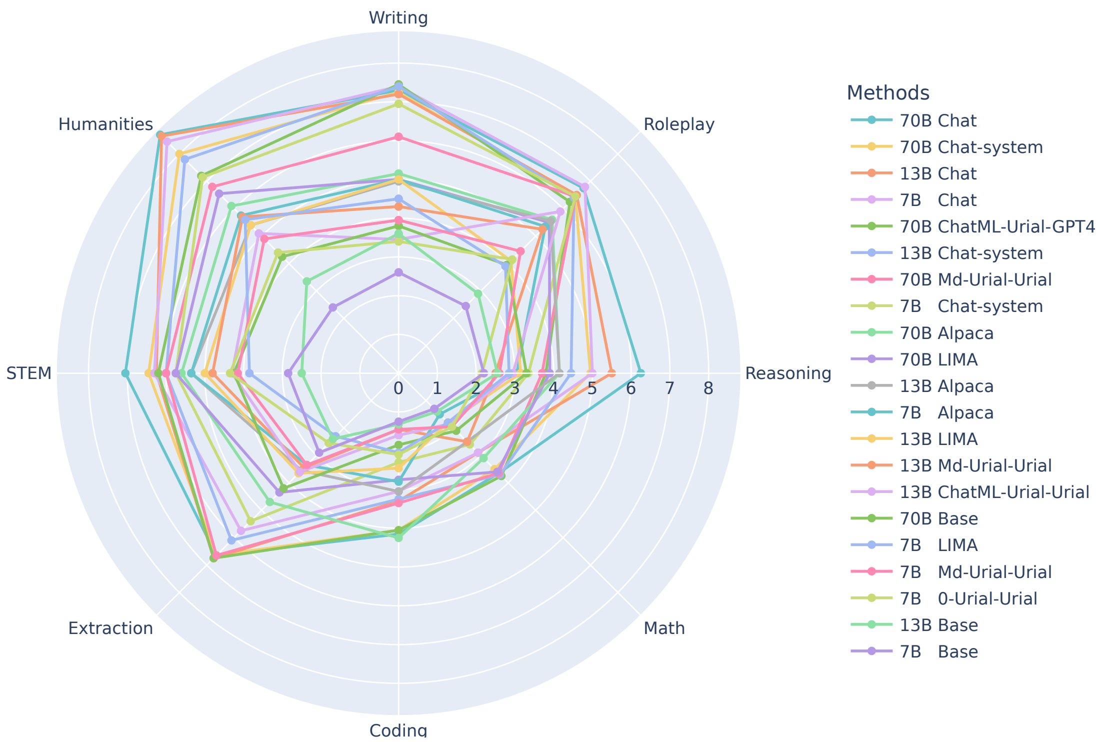

Figure 6: Radar chart of average scores for MT-Bench.