Is In-Context Learning Sufficient for Instruction Following in LLMs?

Hao Zhao **Maksym Andriushchenko** **Francesco Croce** **Nicolas Flammarion**
Theory of Machine Learning Lab
EPFL, Lausanne, Switzerland
{hao.zhao, maksym.andriushchenko, francesco.croce, nicolas.flammarion}@epfl.ch

Abstract

In-context learning (ICL) allows LLMs to learn from examples without changing their weights: this is a particularly promising capability for *long-context* LLMs that can potentially learn from *many* examples. Recently, Lin et al. (2024) proposed URIAL, a method using only three in-context examples to align base LLMs, achieving non-trivial instruction following performance. In this work, we show that, while effective, ICL alignment with URIAL still underperforms compared to instruction fine-tuning on the established benchmark MT-Bench, especially with more capable base LLMs. We then uncover the most relevant elements for successful in-context alignment, finding the crucial role of the decoding parameters. Based on these insights, we show that the approach of URIAL can indeed be improved by adding *high-quality*, possibly carefully selected via greedy search, demonstrations in context, getting closer to the performance of instruct models. Finally, we provide the first, to our knowledge, systematic comparison of ICL and instruction fine-tuning (IFT) for instruction following in the low data regime, where ICL can be a viable alternative to IFT. Overall, our work advances the understanding of ICL as an alignment technique and its relationship to IFT. We provide our code at `https://github.com/tml-epfl/icl-alignment`.

1 Introduction

The large-scale pre-training phase allows Large Language Models (LLMs) to acquire extensive knowledge and capabilities (Bubeck et al., 2023). However, these base models struggle to follow instructions directly from prompts, necessitating further fine-tuning to be used as conversational models. Inspired by Brown et al. (2020) who showed that LLMs can learn from demonstrations provided as part of the input—a concept known as in-context learning (ICL)—Lin et al. (2024) have recently studied the feasibility of *in-context alignment* (Han, 2023; Li et al., 2023b). They found that including merely three carefully selected question-answer pairs in the prompt is sufficient to make *base* models follow instructions and interact with users at a similar level to instruction-fine-tuned models on their own benchmark.

Analysis of in-context alignment. In this work, we extend the evaluation of the URIAL prompt strategy proposed by Lin et al. (2024) across several base models, including GPT-4-Base,[1] and on established instruction following benchmark MT-Bench (Zheng et al., 2023), see Table 2. First, we show that, although URIAL achieves reasonable performance, it still lags behind instruction-fine-tuned models. Then, to better understand the success but also weaknesses of in-context alignment, we analyze which are the most relevant ingredients in URIAL. We find that the decoding parameters in the LLMs generation (temperature, sampling scheme, etc.) may crucially influence the performance, and the optimal configuration allows even base models to achieve good scores on MT-Bench.

[1]We received access to the base GPT-4 model via the OpenAI Researcher Access Program.

NeurIPS 2024 Workshop on Adaptive Foundation Models.

Many-shot in-context alignment. Then, we test various strategies to improve in-context alignment, leveraging recent models with *extensive context windows* which allow for longer in-context prompts. In particular, we study the effect of *many-shot* in-context learning by adding *high-quality* demonstrations from existing instruction datasets. Unlike what suggested by Lin et al. (2024), this approach can improve upon URIAL, but only when using high-quality examples, as those from SkillMix. However, it is still not sufficient to fully close the gap with aligned LLMs, as the performance plateaus after 10-30 in-context examples. This behavior is in contrast to many-shot ICL for tasks like summarization (Narayan et al., 2018), translation (Costa-jussà et al., 2022), or classification (Li et al., 2024), where providing many examples is clearly beneficial (Agarwal et al., 2024; Bertsch et al., 2024). We further test a simple greedy algorithm to select the in-context examples which optimize the MT-Bench score. This selection scheme outperforms, with 1 to 3 additional demonstrations, the many-shot approach with random samples, and allows to further reduce the distance from in-context aligned models to fine-tuned models.

ICL vs IFT for alignment. While our experiments suggest that ICL cannot match the instruction-following performance of models aligned through costly fine-tuning, possibly using preference data, it remains an open question whether ICL can compete with or outperform IFT in the low-data regime. We provide an extensive evaluation on multiple LLMs and datasets of both approaches when varying the question-answer dataset size between 3 and 4000 examples. We observe that, with high-quality data, ICL and IFT achieve almost identical 1st-turn MT-Bench score. Surprisingly, in the 2nd-turn score, IFT clearly outperforms ICL, with ICL performing even worse than the base model. This suggests that ICL suffers from some heavy overfitting to the style of the examples shown in context.

Overall, these results, complemented by several ablation studies, provide a more nuanced picture of ICL as an alignment technique compared to previous works. Moreover, our comparison of ICL to IFT when using the same data bridges a gap in the literature about understanding these orthogonal approaches for adapting pre-trained LLMs into conversational models.

2 Uncovering the Limits of ICL for Instruction Following

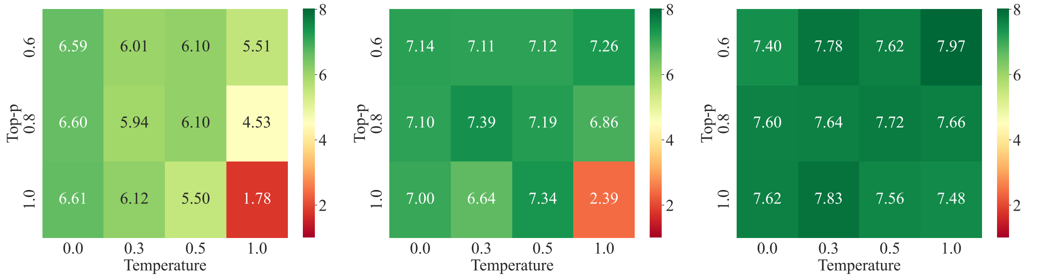
In the following, we provide an in-depth analysis which aims at (1) systematically comparing the performance of base models plus URIAL to that of aligned models (see Sec. 2.1), (2) understanding which components of URIAL (and in-context alignment in general) are most important for its effectiveness (Sec. 2.2), (3) testing the influence of question-answers format in our task (App. B).

2.1 Systematic evaluation of URIAL

We compare the performance of several base models with the URIAL in-context prompt to that of their instruction fine-tuned versions. Table 2 shows the results on MT-Bench (Zheng et al., 2023), one of the most popular benchmarks for instruction following ability of LLMs. We compare several LLMs of different sizes and capabilities, ranging from Llama-2-7B (Touvron et al., 2023) to the *base* GPT-4 model (OpenAI, 2023). We observe that base models with URIAL achieve competitive performance on the 1st-turn score, but cannot match that of their instruction fine-tuned counterparts in all cases except for Llama-2-70B and Mistral-7B-v0.1 (both originally included in Lin et al. (2024), unlike most others). Conversely, the 2nd-turn score with URIAL is *significantly worse* than for instruction-tuned LLMs: we hypothesize that this is because no multi-turn demonstrations are given in context.

2.2 What matters for in-context alignment via URIAL?

Decoding parameters. To clarify the influence of these parameters that were not explicitly discussed in Lin et al. (2024), we compute the performance of base LLM, base LLM with URIAL, and fine-tuned LLM when varying decoding configurations. In Fig. 1 we fix the repetition penalty to 1.15, and vary temperature and top-p, for the Mistral-7B-v0.2 models. Surprisingly, we notice that with the decoding parameters of URIAL (temperature $= 0$, top-p $= 1$) even the base model without any in-context example achieves reasonable MT-Bench score (6.61 vs 7.00 of URIAL). Moreover, the temperature value appears to have the most influence of the performance of the base model. With URIAL, almost all configurations provide very similar results, with the exception of temperature $= 1$, top-p $= 1$ (possibly because it allows the sampling of generated tokens from the tail of the token



(a) Base model (b) Base model with URIAL (c) Instruct model

Figure 1: **Effect of decoding parameters on the 1st-turn MT-Bench scores across models.** We vary temperature and top-p, with a fixed repetition penalty of 1.15 as used in the URIAL codebase. See the complete results in App. E.2.

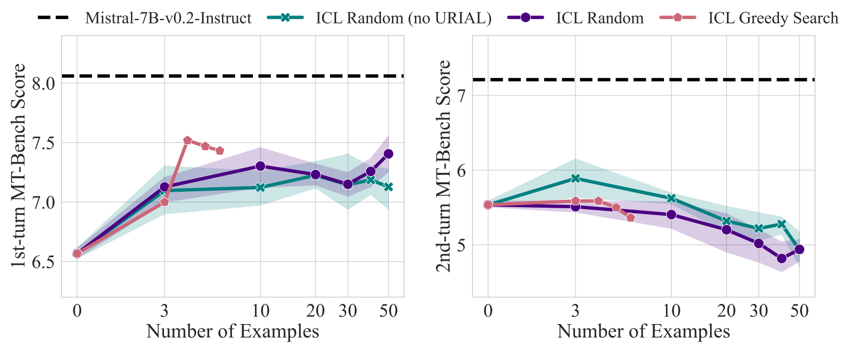


Figure 2: **Scaling the number of demonstrations for alignment with ICL on Mistral-7B-v0.2.** We measure the alignment performance of different settings using the MT-Bench score. See the results on Llama-3.1-8B in Fig. 6.

distribution, thus producing low-quality text). Finally, the aligned model performs similarly across all decoding schemes, with slightly better results than URIAL. These results suggest that (1) the decoding parameters are an *overlooked factor* contributing to the success of in-context alignment, and (2) fine-tuning adjusts the sampling distribution of language models so that even with high-variance decoding configurations the generated text preserves high quality. We provide additional analysis for no repetition penalty (i.e. $= 1$) in Fig. 10 and Llama-3.1-8B in Fig. 11 in App. E.2: these extensive experiments further validate, with different base LLMs and settings, that the decoding scheme is a crucial factor for the instruction-following behavior of base models, and also affects, but to lower degree, in-context alignment.

3 Many-Shot In-Context Learning for Instruction Following

Sec. 2.1 indicate that URIAL alone is not able, in most cases, to reach the performance of instruct models. In the following, we explore whether we can close this gap. We focus on Mistral-7B-v0.2 (Jiang et al., 2023) and Llama-3.1-8B (Dubey et al., 2024) since (a) both URIAL and the aligned model achieve competitive performance in Table 2, and (b) these base models have context windows of 32k and 128k respectively, which can fit many examples (around 50 for Mistral-7B-v0.2 and more than 200 for Llama-3.1-8B). Details on the experimental setup are in App. A.

3.1 Scaling up the number of in-context demonstrations

Given the success of many-shot ICL (Agarwal et al., 2024), we test the effect of increasing the number of in-context demonstrations. We sample random examples from the SkillMix dataset, since it contains high-quality question-answer pairs. We consider two scenarios: (1) all demonstrations are randomly chosen from SkillMix (the set of rules from URIAL is kept), and (2) we add the new demonstrations on top of the URIAL examples. In Fig. 2, we report the results on MT-Bench when varying the number of demonstrations. We repeat sampling with 5 random seeds and show mean and

Table 1: **Improved in-context (IC) alignment by adding to URIAL prompt the demonstrations found via greedy search.** We find the optimal IC prompts in an incremental way by selecting query-response pairs from the high-quality SkillMix dataset. We report the 1st-turn score on MT-Bench and the length-controlled (LC) win rates on AlpacaEval 2.0 when using different IC prompts.

Model	Mistral-7B-v0.2		Llama-3.1-8B	
	MT-Bench (1st)	AlpacaEval 2.0	MT-Bench (1st)	AlpacaEval 2.0
URIAL (3 examples)	7.00	8.22	6.95	7.28
URIAL + greedy search (1 ex.)	**7.52**	7.53	7.61	**8.61**
URIAL + greedy search (2 ex.)	7.47	7.78	7.77	8.16
URIAL + greedy search (3 ex.)	7.43	**8.55**	**7.81**	8.19

standard deviation over the corresponding results. We observe that for both base LLMs, the 1st-turn score increases with 3, 10 and 20 examples, but then plateaus without clear benefits from scaling up beyond 30 demonstrations (for Llama-3.1-8B the trend becomes even slightly negative with more than 100 examples). Thus, we find that adding up to 30 high-quality examples from SkillMix can improve alignment via ICL. This result contrasts with the findings of Lin et al. (2024), which showed that URIAL performed better with 3 demonstrations than with 8.

Next, we observe that in-context alignment does not improve the second-turn score. In fact, adding more examples continues to decrease performance, even falling below that of the base models. This behavior is likely due to the presence of only single-turn examples in the prompt, causing the LLM to respond in the same style without adapting to more complex conversations. Overall, these results show that, unlike in the setting of Agarwal et al. (2024); Bertsch et al. (2024), simply scaling the number of ICL examples is not sufficient to consistently improve the instruction following performance.

Finally, we notice that on Llama-3.1-8B, using three examples from SkillMix attain, on average, better performance than using the three examples of URIAL. Overall, there is no significant difference in the scaling behavior between using or not using the URIAL demonstrations. Thus, we conclude that the success of in-context alignment is only partially dependent on the demonstrations themselves, provided they are of sufficient quality (see also Sec. 4).

3.2 Greedy search for effective demonstrations

Given the limited success of adding random demonstrations to URIAL, we try to greedily maximize the MT-Bench score by testing 100 high quality instructions from SkillMix as the 4th additional example to URIAL. For each resulting ICL prompt, we compute the MT-Bench score with GPT-4-Turbo as judge instead of GPT-4 (used in MT-Bench) as the former is faster, cheaper, and helps mitigate potential overfitting to the benchmark score. We then repeat this procedure sequentially to find a 5th and a 6th demonstration, restricting the search space at each round to only instructions leading to a high enough MT-Bench score to reduce the computational cost (see details in App. A.3).

We add the results of the greedy search to the plots in Fig. 2, computing the true MT-Bench (i.e., with GPT-4 as judge). For both base models, the 4th example found by greedy search is sufficient to match the best 1st-turn score achieved by scaling the in-context examples with random demonstrations (see Sec. 3.1). In Table 1, we provide the details of these results: the 4th ICL example yields a significant improvement over URIAL, e.g., from 6.95 to 7.61 for Llama-3.1-8B, with only a further +0.20 given by the 5th and 6th. Overall, this evaluation further indicates that the improvement in instruction following one can achieve with in-context alignment quickly saturates when increasing the number of examples. In Fig. 5, we show the distribution of the scores (with GPT-4-Turbo as judge) at step of the greedy search when using Llama-3.1-8B as base model (the analog for Mistral-7B-v0.2 in App. E.1). Most demonstrations improve the score when added as the 4th example on top of URIAL (as shown in the first plot), but no further progress is observed with additional demonstrations.

We further test the IC prompts found by the greedy search on the AlpacaEval 2.0 (Li et al., 2023a) benchmark, where we measure length-controlled win rate. As shown in Table 1, using the examples given by greedy search leads to better results than plain URIAL on AlpacaEval 2.0, without directly optimizing for it. This improvement indicates that the found IC prompt does not completely overfit to the MT-Bench score, although the gains are less consistent than on MT-Bench.

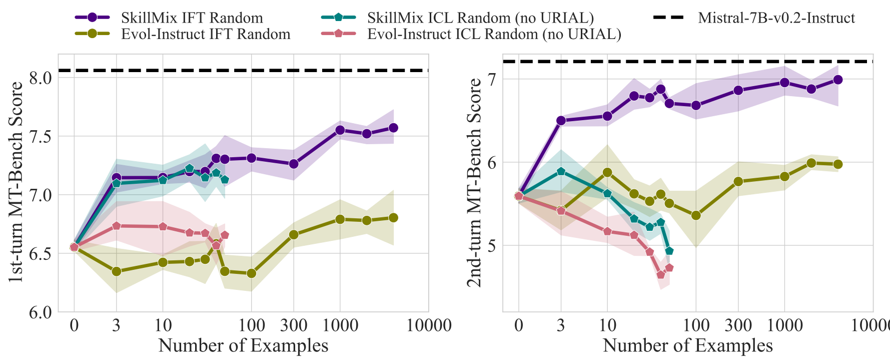


Figure 3: **Comparison of ICL vs IFT for alignment in the low data regime.** We measure the alignment performance of different settings for Mistral-7B-v0.2 using the MT-Bench score. See the results on Llama-3.1-8B in Fig. 7.

4 A Comparison of ICL vs IFT for Instruction Following

The results from the previous sections strongly suggest that in-context alignment might not be as competitive as sophisticated, computationally heavy alignment techniques. However, it is unclear how it compares to more lightweight approaches, such as IFT, when applied to a small number of examples. In the low-data regime, the choice between ICL and IFT can be viewed as a decision between allocating resources for fine-tuning (which permanently modifies the model's weights) or increasing inference time by adding the in-context prompt each time. The description of experiment setups is shown in App. A.4.

Results on Mistral-7B-v0.2. Fig. 3 shows that IFT and ICL on SkillMix (violet and green curves respectively) perform almost identically in 1st-turn score until 50 examples are used. Beyond this point, increasing the number of training examples consistently improves the performance of IFT. The strong performance of IFT in the low-data regime is particularly surprising, as one might expect overfitting when training on very few examples (as few as 3) over several epochs, yet this does not occur. Finally, ICL and IFT show nearly opposite trends for the 2nd-turn score: increasing the dataset size benefits IFT (almost reaching the performance of the instruct model) but detrimental to ICL. This behavior suggests that, while IFT is less flexible due to the model weights being updated, it generalizes better to tasks different from those used for alignment (recall that the instructions in SkillMix are single-turn only). The results on Llama-3.1-8B and the discussion of data quality are shown in App. C.

5 Conclusions

In this work, we have first illustrated that ICL via URIAL is a good baseline for instruction-following alignment, but with a few limitations: it typically performs slightly worse than IFT on single-turn conversations and does not generalize well to multi-round ones. Then, we have uncovered the key components for IC alignment: e.g. the decoding parameters alone can, surprisingly, make base models reasonably good at instruction following. Adding in-context high-quality demonstrations improves performance beyond what previously suggested (Lin et al., 2024), but not enough to reach LLMs aligned with sophisticated methods (e.g., RLHF). We conjecture that via ICL, the LLM can learn to infer the response style, but the overall learning signal is not sufficiently strong to benefit from a large amount of examples, despite the long context windows of recent LLMs.

Moreover, to the best of our knowledge, we have provided the first systematic comparison of ICL and IFT for instruction following when using the same (small) number of demonstrations. Surprisingly, the two approaches are roughly equivalent in terms of single-turn conversation performance. However, models trained via IFT, unlike ICL, can generalize to multi-round conversations, which are out-of-distribution compared to the examples used for alignment. Overall, our work provides a deeper and more complete understanding of how in-context alignment works, as well as of its potential and limitations, in particular in comparison to fine-tuning.

Acknowledgements

We are grateful to OpenAI for providing us API credits and access to GPT-4-Base as part of the OpenAI Researcher Access Program. M.A. was supported by the Google Fellowship and Open Phil AI Fellowship. This work was supported by the Swiss National Science Foundation (grant number 212111) and by an unrestricted gift from Google.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*, 2024.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

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

Simran Kaur, Simon Park, Anirudh Goyal, and Sanjeev Arora. Instruct-skillmix: A powerful pipeline for llm instruction tuning. *arXiv preprint arXiv:2408.14774*, 2024.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023a.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*, 2023b.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=wxJ0eXwwda`.

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

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. WizardLM: Empowering large language models to follow complex instructions. *ICLR*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS Datasets and Benchmarks Track*, 2023.

Table 2: **Systematic comparison of URIAL to aligned models on MT-Bench across different base LLMs.** For several recent model families, we compare the performance on instruction following tasks of the base LLMs plus URIAL (i.e. in-context alignment) to that of the instruct models, fine-tuned with sophisticated techniques like supervised instruction fine-tuning and RLHF. In most cases, the fine-tuned models outperform URIAL. * denotes the result taken from the URIAL GitHub repository.

Model	1st-turn	2nd-turn	Average
Llama-2-7B + URIAL *	5.75	3.91	4.83
Llama-2-7B-Instruct	**7.14**	**5.91**	**6.53**
Llama-2-70B + URIAL *	**7.61**	6.61	7.11
Llama-2-70B-Instruct	7.37	**7.03**	**7.20**
Llama-3-8B + URIAL *	6.84	4.65	5.75
Llama-3-8B-Instruct	**8.29**	**7.42**	**7.86**
Llama-3-70B + URIAL *	7.71	5.09	6.40
Llama-3-70B-Instruct	**8.96**	**8.51**	**8.74**
Llama-3.1-8B + URIAL *	6.95	5.31	6.13
Llama-3.1-8B-Instruct	**8.27**	**7.73**	**8.00**
Mistral-7B-v0.1 + URIAL *	**7.49**	5.86	6.67
Mistral-7B-Instruct-v0.1	7.31	**6.39**	**6.85**
Mistral-7B-v0.2 + URIAL *	6.99	5.55	6.27
Mistral-7B-Instruct-v0.2	**8.06**	**7.21**	**7.64**
Mixtral-8x22B-v0.1-4bit + URIAL	8.28	7.14	7.71
Mixtral-8x22B-Instruct-v0.1-4bit	**8.78**	**8.25**	**8.52**
GPT-4-Base + URIAL	7.96	5.04	6.50
GPT-4 (March 2023) *	**8.96**	**9.03**	**8.99**

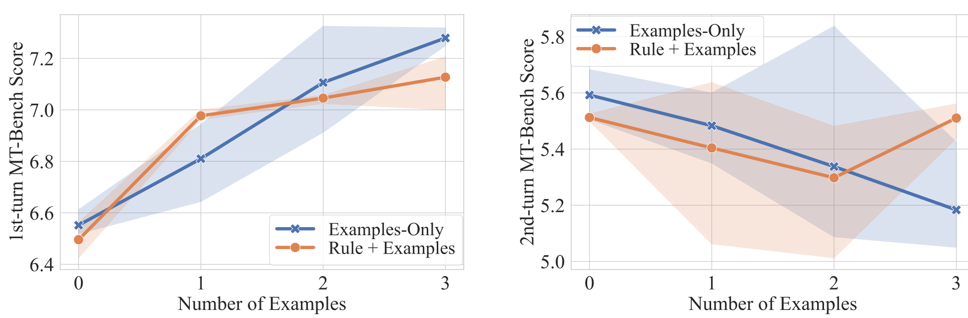


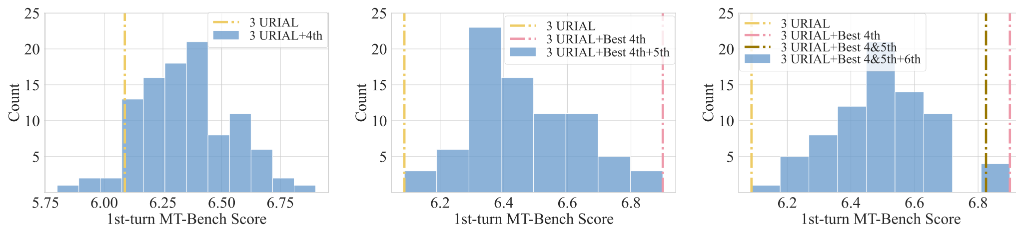
Figure 4: **Influence of the individual components of URIAL.** We report 1st and 2nd turn MT-Bench score for every subset of the three demonstrations of URIAL with Mistral-7B-v0.2. We test each configuration with ("Rule + Examples") and without ("Examples-Only") the URIAL set of rules in the in-context prompt. We observe a clear increasing trend in 1st-turn score with more examples, but a decrease in the 2nd-turn performance. The set of rules does not seem to influence the results. The randomness for 0 and 3 examples is caused by the small fluctuations in the score of the GPT-4 judge.

A Experimental Details

In this section, we show more details about the experiments we conducted in the paper.

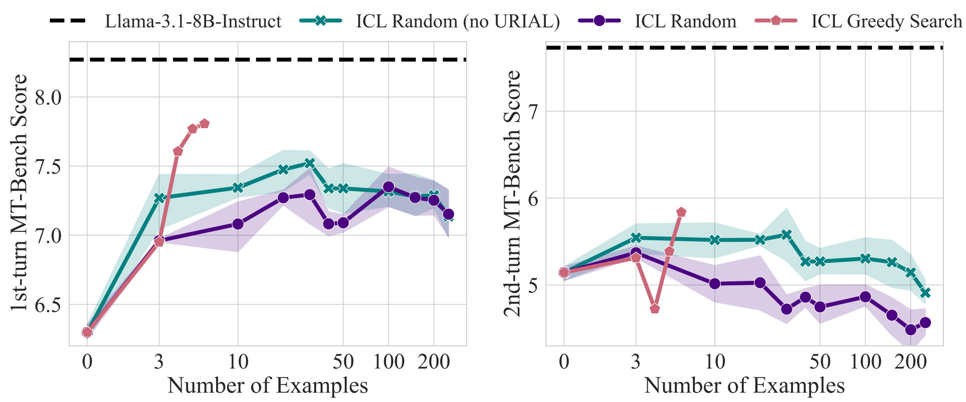
A.1 Datasets and base models

We select instruction-following demonstrations for the base LLMs to learn in context from open-sourced instruction fine-tuning datasets: (a) SkillMix-4k (Kaur et al., 2024) is an automated approach for synthesizing diverse, high-quality IFT data. It primarily involves two stages: first leveraging the LLMs to propose a set of critical "skills" for instruction-following, from which a pair of skills are randomly chosen to facilitate synthetic data generation based on powerful LLMs. (b) Evol-Instruct-70k (Xu et al., 2024) contains 70k training examples with varying complexity and is well-known for



(a) Search for 4th example (b) Search for 5th example (c) Search for 6th example

Figure 5: **The distribution of the 1st-turn MT-Bench score (GPT-4-Turbo as judge) on Llama-3.1-8B obtained by adding multiple instructions from SkillMix (Kaur et al., 2024) as a 4th (a), 5th (b), 6th (c) demonstration to URIAL.** The dashed lines of various colors refer to the 1st-turn MT-Bench score of the obtained searching results. A majority of the 4th examples contribute positively to the model's instruction-following performance, but the improvement quickly diminishes when running the greedy search for 5th and 6th demonstrations.



(a) Llama-3.1-8B

Figure 6: **Scaling the number of demonstrations for alignment with ICL on Llama-3.1-8B.** We measure the alignment performance of different settings using the MT-Bench score. ICL with more demonstrations quickly saturates and does not bridge the performance gap between the base model and its aligned counterpart. In particular, the ICL alignment performance of 3 random examples from the high-quality SkillMix dataset surpasses that of 3 examples from URIAL.

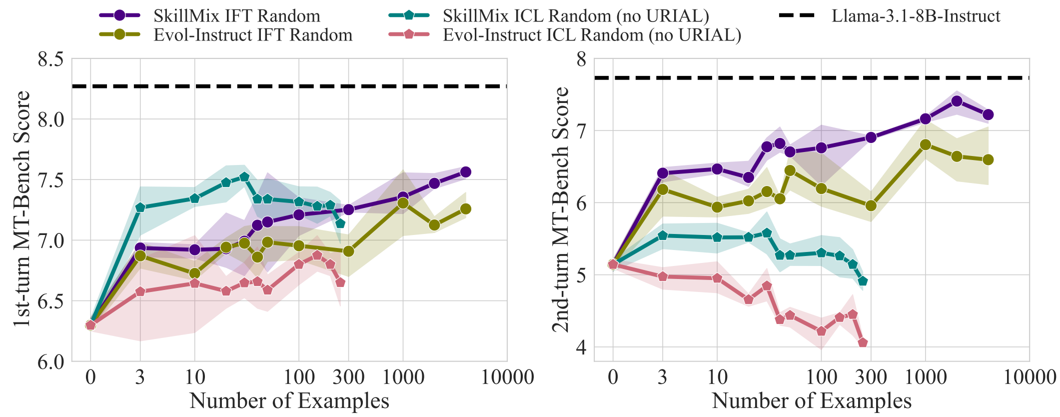
the use to build the series of WizardLM models. SkillMix-4k is the primary dataset for all experiments in the paper and Evol-Instruct-70k is solely used for scaling experiments and ablation study of data quality.

We compare ICL performance across multiple models that have sufficiently large context windows:

1. Mistral-7B-v0.2 (Jiang et al., 2023) has 7.3 billion model parameters. On many established benchmarks, it outperforms LLMs that have significantly more parameters, such as Llama-2-13B and Llama-1-34B. The trained context length of Mistral-7B-v0.2 is 32k tokens.
2. Llama-3.1-8B (Dubey et al., 2024) is the newest and most powerful 8B model from Meta when we are writing, and it supports multilingual dialogue use cases. It supports input texts containing up to 128k tokens.

In some experiments, the following base LLMs are used:

1. Llama-2-7B-80k (Fu et al., 2024) is a variant of Llama-2-7B model fine-tuned with 80k context on a carefully designed long-document data mixture.
2. Mixtral-8x22B-v0.1-4bit is a variant of Mixtral-8x22B-v0.1 (Jiang et al., 2024), which is a pre-trained generative sparse Mixture of Experts, quantized with 4-bit precision. It contains $\sim$176B parameters and $\sim$44B active during inference, and it has a 65k context window.



(a) Llama-3.1-8B

Figure 7: **Comparison of ICL vs IFT for alignment in the low data regime.** We measure the alignment performance of different settings for Llama-3.1-8B using the MT-Bench score. IFT with more demonstrations keeps improving the alignment performance, almost bridging the gap between the base model and its aligned counterpart. IFT-aligned models perform well on multi-turn conversations, unlike with ICL. Finally, data quality has significant impact on both IFT and ICL: the higher-quality SkillMix leads to better performance than Evol-Instruct.

Table 3: **Importance of question-answer matching of demonstrations for in-context alignment.** We report the 1st-turn MT-Bench score of Mistral-7B-v0.2 and Llama-3.1-8B when varying the structure of the in-context examples $\{(X_i, Y_i)\}_i$, where X_i is the query and Y_i is the corresponding ground-truth answer.

In-context prompt	URIAL (3 examples)		URIAL + Greedy Search (6 examples)	
	Mistral-7B-v0.2	Llama-3.1-8B	Mistral-7B-v0.2	Llama-3.1-8B
no demonstrations	6.52	6.25	6.52	6.25
X only (no Y)	5.90	4.48	5.53	5.10
Y only (no X)	<u>6.94</u>	<u>6.83</u>	<u>7.02</u>	<u>6.98</u>
circular shift of Y	5.04	5.69	5.59	2.13
in-domain random Y	6.24	6.26	4.63	5.49
out-of-domain random Y	4.13	3.73	1.50	4.36
correct Y	**6.99**	**6.95**	**7.43**	**7.81**

We use the same decoding configuration as what is used in URIAL (Lin et al., 2024). Concretely, we employ greedy decoding, i.e., $temperature = 1.0$, for all models, including base and instruction fine-tuned models, to maximize reproducibility and secure a fair and robust evaluation. $Top - p = 1.0$ is adopted to keep the full cumulative probability distribution. Besides, we use $repetitionpenalty = 1.15$[2] on base models to prevent degeneration.

A.2 Evaluation

MT-Bench (Zheng et al., 2023), consists of 80 high-quality and challenging questions that have two-round interaction, designed to examine the multi-turn conversation and instruction-following capability of models. It features 8 common categories of user prompts: coding, math, reasoning, extraction, roleplay, writing, humanities/social science, and STEM.

AlpacaEval 2.0 (Li et al., 2023a) provides 805 test instructions, on which we generate new responses using the target model, and then calculate the score by competing with the baseline model (i.e., GPT-4-Turbo) judged by a designated automatic evaluator. We apply the AlpacaEval 2.0 benchmark

[2] As in the codebase at `https://github.com/Re-Align/URIAL/blob/main/run_scripts/mt-bench/_run_mt_bench.sh`.

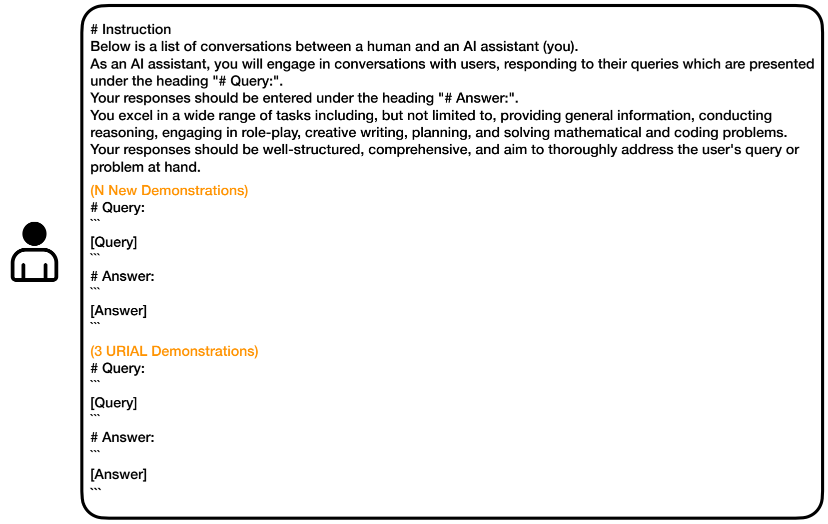


Figure 8: The prompt template for doing ICL in our work. Note that the words in orange are solely for illustration purposes and do not appear in the real prompts. The mixture of new demonstrations we test in our experiments is inserted before URIAL demonstrations.

in our experiments to ensure that the effective demonstrations found through greedy search don't overfit to MT-Bench.

A.3 Greedy search

Firstly, we randomly sample 100 examples from the high-quality IFT dataset, SkillMix-4k, and then create 100 new 4-shot prompts by adding each one of the 100 sampled examples, as the fourth demonstration, to the original 3-shot URIAL prompt. We evaluate the resulting instruction-following performance for each prompt on MT-Bench but with GPT-4-Turbo as the judge since GPT-4-Turbo is cheaper and faster than GPT-4 (the original LLM judge for MT-Bench evaluation), and also has a high correlation with human judgment. Finally, the example with the highest 1st-turn MT-Bench score is chosen.

We continue our greedy search for the proper fifth and sixth demonstration in a more restricted search space due to the heavy computational cost of the search process. We keep reducing the search space by applying a threshold of 1st-turn MT-Bench score (i.e., 6.2 in our experiments). Similarly, we add each example from the new search space on top of the best (N-1)-shot prompt found in previous steps and generate a set of new N-shot prompts. Then we run the MT-bench evaluation with GPT-4-Turbo as the judge for multiple times and select the best (N+1)-shot prompt.

A.4 Scaling experiments

The recently released LLMs with increasingly large context windows allow using many more shots for ICL than are typically used. Prior works Agarwal et al. (2024); Bertsch et al. (2024) have shown that many-shot ICL, compared to few-shot ICL, can make task-specific fine-tuning less essential and allows LLMs to tackle a wider range of tasks without specification. Therefore, we want to examine if the instruction-following performance of base LLMs could benefit from many-shot ICL through various scaling experiments, and systematically compare ICL with IFT. In the following, we add more details about these scaling experiments, including the prompt template, data selection for both ICL and IFT, and training hyper-parameters for IFT experiments.

ICL Random. We construct the set of in-context demonstrations based on the high-quality data we select from SkillMix-4k or Evol-Instruct-70k dataset. Through randomly sampling from the high-quality dataset for multiple times, we generate a series of in-context demonstration sets that each contains N examples, where $N \in \{0, 7, 17, 27, 37, 47\}$ for Mistral-7B-v0.2 model (32k context length) and $N \in \{0, 7, 17, 27, 37, 47, 97, 147, 197, 247\}$ for Llama-3.1-8B model (128k context length), and insert the N demonstrations into the prompt template for ICL as shown in Fig 8. Note that the prompt template is strictly following the one used in URIAL paper and we only replace the demonstrations used for the ICL purpose. The average performance and standard deviations are computed over 5 random seeds.

ICL Random (no URIAL). We generate a series of in-context demonstration sets by randomly sampling from the high-quality IFT dataset, and each set contains N examples, where $N \in \{3, 10, 20, 30, 40, 50\}$ for Mistral-7B-v0.2 model (32k context length) and $N \in \{3, 10, 20, 30, 40, 50, 100, 150, 200, 250\}$ for Llama-3.1-8B model (128k context length). URIAL examples will not be added to the context, so it ensures the total number of in-context examples are the same as the Random group. The average performance and standard deviations are computed over 5 random seeds.

ICL Greedy Search. Following the procedure we mention in Sec. A.3, ideally we can get as many optimal examples as we want if we have sufficient OpenAI API credits. Thus it allows us to create another series of in-context demonstration sets by selecting another N examples for each $N \in \{1, 2, 3\}$ in the paper. The resulting mixture of in-context demonstrations is then placed in the corresponding location of the prompt template as shown in Fig. 8.

Table 4: Details of training hyperparameters for IFT experiments.

Data Size	# GPUs	Epochs	LR	LR Scheduler	Batch Size	Context Win. Len.	WD	Warmup Rate
Mistral-7B-v0.2								
3	2	6	2e-6	Cosine	2	2048	0.01	0.03
10	4	6	2e-6	Cosine	8	2048	0.01	0.03
20	4	6	2e-6	Cosine	8	2048	0.01	0.03
30	4	6	2e-6	Cosine	8	2048	0.01	0.03
40	4	6	2e-6	Cosine	8	2048	0.01	0.03
50	4	6	2e-6	Cosine	8	2048	0.01	0.03
100	4	6	2e-6	Cosine	8	2048	0.01	0.03
300	4	6	2e-6	Cosine	8	2048	0.01	0.03
1000	4	15	2e-6	Cosine	128	2048	0.01	0.03
2000	4	15	2e-6	Cosine	128	2048	0.01	0.03
4000	4	15	2e-6	Cosine	128	2048	0.01	0.03
Llama-3.1-8B								
3	2	6	4e-6	Cosine	2	2048	0.01	0.03
10	4	6	4e-6	Cosine	8	2048	0.01	0.03
20	4	6	4e-6	Cosine	8	2048	0.01	0.03
30	4	6	4e-6	Cosine	8	2048	0.01	0.03
40	4	6	4e-6	Cosine	8	2048	0.01	0.03
50	4	6	4e-6	Cosine	8	2048	0.01	0.03
100	4	6	4e-6	Cosine	8	2048	0.01	0.03
300	4	6	4e-6	Cosine	8	2048	0.01	0.03
1000	4	15	4e-6	Cosine	128	2048	0.01	0.03
2000	4	15	4e-6	Cosine	128	2048	0.01	0.03
4000	4	15	4e-6	Cosine	128	2048	0.01	0.03

IFT. We run instruction fine-tuning the base LLMs (Mistral-7B-v0.2 and Llama-3.1-8B) either with tens of examples (i.e., few-sample regime) or thousands of examples. The training examples are randomly sampled from existing IFT datasets, SkillMix-4k and Evol-Instruct-70k. The mean score and standard deviations are calculated over multiple random seeds. More specifically, we use 5 random seeds for Mistral-7B-v0.2 model and 3 random seeds for Llama-3.1-8B model due to a restricted compute budget. In particular, since the size of SkillMix dataset is 4k, the randomness of IFT with 4k examples primarily comes from optimization process and scoring evaluation with GPT-4, otherwise part of randomness also comes from data sampling process. We list the details of training hyper-parameters used in IFT experiments in Table. 4. We always select the last model checkpoint to run evaluation.

Table 5: **Improved in-context demonstrations selected from Evol-Instruct-70k dataset for Mistral-7B-v0.2 base model.** We report the 1st-turn score on MT-Bench and the length-controlled (LC) win rates on AlpacaEval 2.0 when using different IC prompts.

Model	MT-Bench (1st)	AlpacaEval 2.0
URIAL (3 examples)	6.99	8.09
URIAL + greedy search (1 ex.)	7.46	7.91
URIAL + greedy search (2 ex.)	**7.69**	8.38
URIAL + greedy search (3 ex.)	<u>7.68</u>	9.22

B Importance of question-answer matching

Surprisingly, Min et al. (2022) showed that using demonstrations with *random labels* does not significantly impair the results of ICL on classification and multiple choice tasks. We conduct a similar study for instruction following, see Table 3. Denoting $\{(X_i, Y_i)\}_i$ the set of in-context examples, with X_i the query and Y_i the corresponding ground-truth answer, we test several configurations on two sets of $\{(X_i, Y_i)\}_i$: the 3 URIAL examples and the 3 URIAL examples with the 3 examples found by our greedy search from Sec. 3.2 (to check the effect of increasing the number of demonstrations).

First, we do not use any demonstration, i.e., the original base models: with the decoding parameters found in the previous section, this already achieves reasonable results. Surprisingly, providing *the questions without the answers* (second row in Table 3) degrades the performance, while the opposite, *answers only*, is effective (0.4-0.7 higher scores than the base model). Next, we permute the answers Y_is with a circular shift of one position, so that all correct answers are still contained in the prompt but matched with the wrong question: this significantly degrades the performance, especially with more examples (URIAL + Greedy Search), e.g. Llama-3.1-8B attains a score of only 2.13 (note that the minimum score is 1). Also, for each question, we sample a new answer from those provided for other instructions in the same category (*in-domain*): e.g., a question about coding is paired with an answer from a different coding question, ensuring that, while the content may be incorrect, the style remains appropriate. Although worse than the original one, this configuration achieves decent scores. Finally, we sample answers from instructions belonging to different *out-of-domain* categories, so that even their style does not match what expected for each question: this leads to the worst performance.

These results show that not all the conclusions from Min et al. (2022) apply to ICL used for instruction following. Most importantly, using answers with correct content and, especially, correct style is crucial for the success of ICL. This property becomes even more evident when increasing the number of examples (to 6 instead of 3), and motivates us to use a highly-curated instruction dataset like SkillMix in experiments in the next sections. Finally, this result suggests that ICL for instruction following works differently compared to other less open-ended tasks.

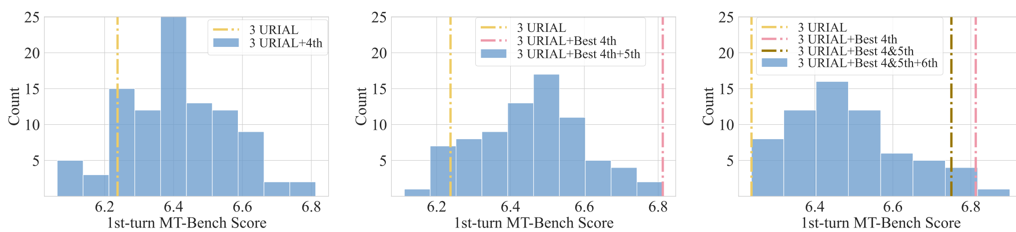
C A comparison of ICL vs IFT on Llama-3.1-8B

Results on Llama-3.1-7B. As shown in Fig. 7, ICL outperforms IFT when using between 3 and 30 demonstrations, though IFT remains effective even with these few examples on this base model. However, we note that IFT matches or exceeds the best performance of ICL (obtained with 20-30 examples) only when using 2k or 4k examples (the entire SkillMix), which represents two orders of magnitude more data. This result confirms that ICL with high-quality data is a viable alternative to IFT when only a limited number of demonstrations are available. Finally, the observations for the 2nd-turn scores are consistent with those for Mistral-7B-v0.2, with IFT consistently outperforming ICL.

Effect of data quality. Finally, we test the effect of using lower-quality instructions compared to SkillMix. In this experiment, we repeat the ICL vs. IFT comparison using random demonstrations from Evol-Instruct-70k (Xu et al., 2024), and show the results in Fig. 3. With Evol-Instruct data, both ICL (red curve) and IFT (yellow curve) perform significantly worse than their counterparts using SkillMix. This trend is consistent across the number of examples, base models, and for both single-turn and multi-turn instructions. Interestingly, the performance gap between IFT on SkillMix and Evol-Instruct is significantly smaller on Llama-3.1-8B than on Mistral-7B-v0.2, suggesting that

Table 6: **Transferability of greedy search IC prompt to other LLMs.** We use the 1st-turn score on MT-Bench and the length-controlled (LC) win rates on AlpacaEval (AE) 2.0 when using various IC prompts on different base models.

Model	MT-B (1st)	AE 2.0
Base model: Llama-2-7B-80k		
URIAL (3 examples)	5.19	1.81
URIAL + greedy search (1 ex.)	5.56	1.50
URIAL + greedy search (2 ex.)	**5.57**	1.84
URIAL + greedy search (3 ex.)	5.10	**2.91**
Base model: Mixtral-8x22B-v0.1-4bit		
URIAL (3 examples)	8.28	14.77
URIAL + greedy search (1 ex.)	8.36	**15.74**
URIAL + greedy search (2 ex.)	7.79	13.20
URIAL + greedy search (3 ex.)	**8.58**	14.68



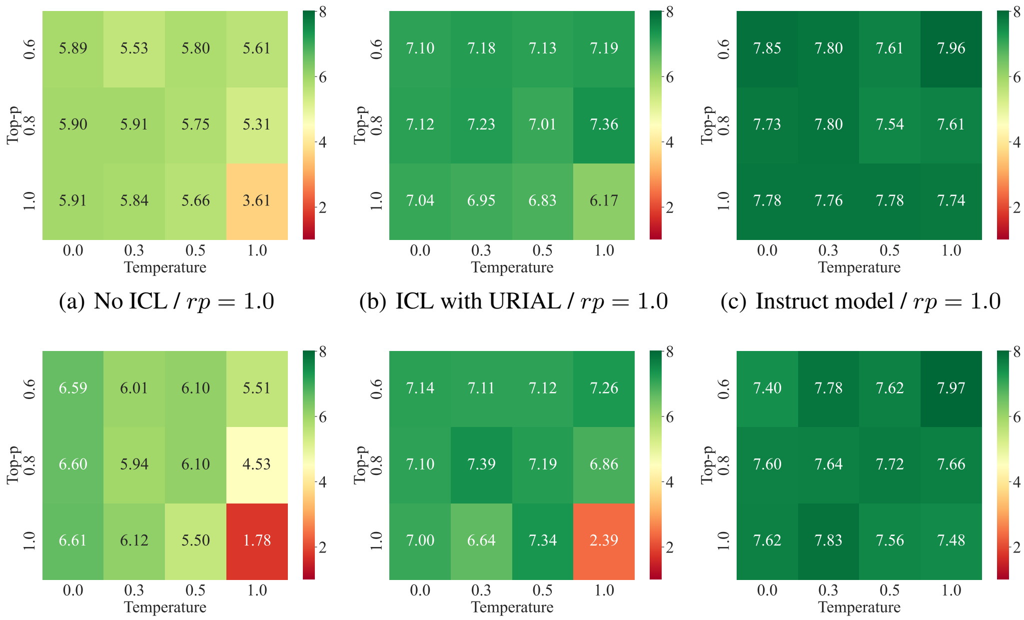
(a) Search for 4th example (b) Search for 5th example (c) Search for 6th example

Figure 9: **The distribution of the 1st-turn MT-Bench score (GPT-4-Turbo as judge) on Mistral-7B-v0.2 obtained by adding multiple instructions from SkillMix (Kaur et al., 2024) as a 4th (a); 5th (b); 6th (c) demonstration to URIAL.** The dashed lines of various colors refer to the 1st-turn MT-Bench score of the obtained searching results. A majority of the 4th examples have positive contributions to the model's instruction-following performance, but the improvement quickly diminishes when running the greedy search for more optimal demonstrations.

better pre-trained models may partially compensate for lower-quality fine-tuning data. Finally, we notice that ICL with 3 examples from Evol-Instruct gets worse 1st-turn scores than URIAL (with also 3 examples), whereas ICL with 3 examples from SkillMix matches (on Mistral-7B-v0.2) or outperforms (on Llama-3.1-8B) URIAL (see also discussion in Sec. 3.1). These observations further confirm the importance of instruction quality for alignment, whether in the context of in-context learning or instruction fine-tuning.

D Transferability of In-Context Prompts

In Table 6, we report the performance of applying the in-context examples found by greedy search (added to URIAL) on Mistral-7B-v0.2 and Evol-Instruct-70k dataset (see Table. 5) to Llama-2-7B-80k (Fu et al., 2024) and Mixtral-8x22B-v0.1-4bit (Jiang et al., 2024). Adding the new examples does not provide a consistent improvement: while it can increase the MT-Bench score (+0.47 on Llama-2-7B-80k, +0.30 on Mixtral-8x22B-v0.1-4bit), it can also, in some cases, give worse results than the original URIAL. Similarly, it yields mixed results when measured by win rate on AlpacaEval 2.0, which was not optimized by the greedy search. This analysis suggests that the most effective ICL demonstrations might vary across base LLMs, potentially because of differences in their pre-training. This could further explain why URIAL underperforms on multiple models in Table 2, especially on those which have become available only recently and were not used for selecting the URIAL examples.



(a) No ICL / $rp = 1.0$ (b) ICL with URIAL / $rp = 1.0$ (c) Instruct model / $rp = 1.0$

(d) No ICL / $rp = 1.15$ (e) ICL with URIAL / $rp = 1.15$ (f) Instruct model / $rp = 1.15$

Figure 10: **The 1st-turn MT-Bench scores of model generations with and without ICL across different decoding schemes.** We mainly consider two hyper-parameters: temperature and top-p. The heatmaps show the answering quality of the base model with and without URIAL in the context is sensitive to the setups of decoding schemes. Surprisingly, with proper decoding parameters, the base model alone is already capable of following instructions.

E Additional experiments

E.1 Greedy Search on Mistral-7B-v0.2

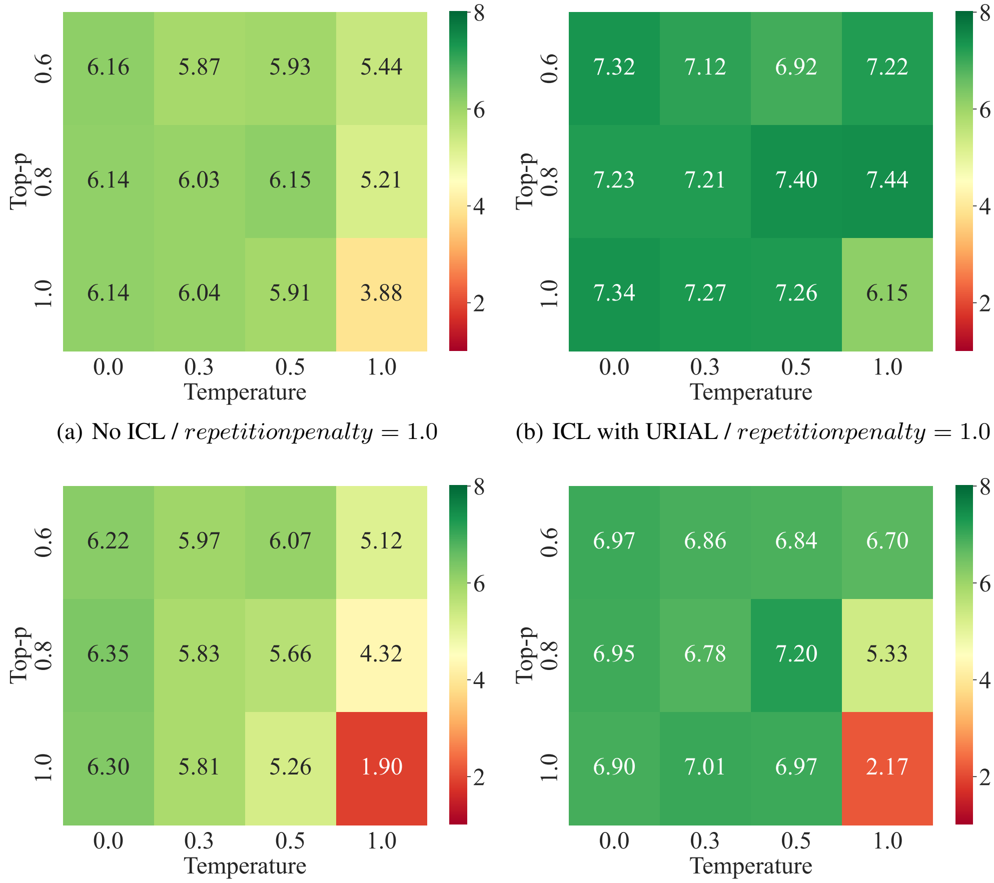
In this section, we show the distribution of the resulting scores (with GPT-4-Turbo as the judge) for each step of the greedy search on SkillMix-4k dataset when using Mistral-7B-v0.2 as the base model. Similar to the finding on Llama-3.1-8B model (see Sec. 3.2), the majority of demonstrations increase the alignment performance, as measured by 1st-turn MT-Bench score, when added as the fourth example on top of URIAL, and few progress is made with more demonstrations. We show more details of greedy search results on Mistral-7B-v0.2 in Fig. 9.

E.2 More experiments on decoding schemes

We show more results of decoding schemes under different settings in this section, as supplementary to the main results in Sec. 2.2. Specifically, in Fig. 10, we turn off the repetition penalty (i.e., $repetition penalty = 1.0$) and show the 1st-turn MT-Bench scores on Mistral-7B-v0.2 when varying the values of temperature and top-p. Moreover, we repeat the same experiment procedure but on a different base model, Llama-3-8B, and show heatmaps in Fig. 11.

E.3 A Breakdown examination of URIAL-aligned models

The test set of MT-Bench (Zheng et al., 2023) is comprised of high-quality questions that belong to 8 common categories: coding, math, reasoning, extraction, roleplay, writing, humanities/social science, and STEM. In Table 7, we present the per-category performance on MT-Bench for each model in order to have a more detailed comparison between the URIAL-aligned models and the corresponding instruct models. The results suggest fine-tuning is almost always better than URIAL (ICL) and only underperforms in a minority of cases on particular base models, such as Mistral-7B-v0.1 and Llama-2-70B.



(a) No ICL / $repetitionpenalty = 1.0$ (b) ICL with URIAL / $repetitionpenalty = 1.0$

(c) No ICL / $repetitionpenalty = 1.15$ (d) ICL with URIAL / $repetitionpenalty = 1.15$

Figure 11: **The 1st-turn MT-Bench scores of Llama-3.1-8B generations with and without URIAL in the context across different decoding schemes.** We mainly consider two hyper-parameters: temperature and top-p.

Table 7: **A breakdown examination of URIAL-aligned models and corresponding instruct models across different categories on MT-Bench.** * denotes the results taken from the URIAL GitHub repository.

Model	coding	math	reasoning	extraction	humanities	roleplay	stem	writing	Average
Llama-2-7B + URIAL *	1.65	1.60	3.45	3.40	8.08	7.48	6.80	6.20	4.83
Llama-2-7B-Instruct	**2.95**	**2.40**	**5.20**	**6.33**	**9.58**	**7.83**	**8.88**	**9.05**	**6.53**
Llama-2-70B + URIAL *	**4.15**	3.60	**6.10**	**7.70**	9.75	7.33	8.75	**9.50**	7.11
Llama-2-70B-Instruct	3.75	**4.10**	5.95	7.40	**9.85**	**7.90**	**9.13**	**9.50**	**7.20**
Llama-3-8B + URIAL *	4.15	2.60	3.50	5.25	8.90	7.30	8.15	6.13	5.75
Llama-3-8B-Instruct	**5.95**	**5.05**	**6.15**	**9.16**	**9.90**	**9.05**	**8.95**	**8.70**	**7.86**
Llama-3-70B + URIAL *	4.35	3.80	4.95	6.20	8.00	7.25	8.55	8.10	6.40
Llama-3-70B-Instruct	**7.85**	**7.35**	**6.25**	**9.75**	**10.00**	**9.30**	**9.60**	**9.80**	**8.74**
Llama-3.1-8B + URIAL *	4.35	3.25	3.95	5.95	9.00	6.95	8.00	7.60	6.13
Llama-3.1-8B-Instruct	**6.40**	**6.50**	**5.70**	**8.78**	**9.80**	**9.00**	**8.60**	**9.20**	**8.00**
Mistral-7B-v0.1 + URIAL *	**4.60**	3.40	4.90	**7.75**	9.08	**7.65**	**8.28**	7.75	6.67
Mistral-7B-Instruct-v0.1	4.35	**3.95**	**6.30**	6.75	**9.45**	7.45	7.70	**8.85**	**6.85**
Mistral-7B-v0.2 + URIAL *	3.80	3.35	4.50	7.45	8.95	6.70	7.43	7.98	6.27
Mistral-7B-Instruct-v0.2	**5.45**	**3.40**	**6.50**	**8.50**	**9.90**	**8.65**	**9.30**	**9.40**	**7.64**
Mixtral-8x22B-v0.1-4bit + URIAL	5.25	5.85	6.00	**9.20**	**9.80**	8.65	8.30	8.60	7.71
Mixtral-8x22B-Instruct-v0.1-4bit	**7.10**	**7.03**	**7.00**	9.10	9.65	**8.90**	**9.65**	**9.70**	**8.52**
GPT-4-Base + URIAL	5.55	5.98	6.90	8.20	5.93	6.90	6.45	6.10	6.50
GPT-4 (March 2023) *	**8.55**	**6.80**	**9.00**	**9.38**	**9.95**	**8.90**	**9.70**	**9.65**	**8.99**