# Conifer: Improving Complex Constrained Instruction-Following Ability of Large Language Models

**Haoran Sun***, **Lixin Liu***, **Junjie Li**, **Fengyu Wang**,
**Baohua Dong**, **Ran Lin**, **Ruohui Huang**
Alibaba Group, Beijing, China
sunhaoran0402@gmail.com, llx271805@alibaba-inc.com

## Abstract

The ability of large language models (LLMs) to follow instructions is crucial to real-world applications. Despite recent advances, several studies have highlighted that LLMs struggle when faced with challenging instructions, especially those that include complex constraints, hindering their effectiveness in various tasks. To address this challenge, we introduce **Conifer**, a novel instruction tuning dataset, designed to enhance LLMs to follow multi-level instructions with complex constraints. Utilizing GPT-4, we curate the dataset by a series of LLM-driven refinement processes to ensure high quality. We also propose a progressive learning scheme that emphasizes an easy-to-hard progression, and learning from process feedback. Models trained with `Conifer` exhibit remarkable improvements in instruction-following abilities, especially for instructions with complex constraints. On several instruction-following benchmarks, our 7B model outperforms the state-of-the-art open-source 7B models, even exceeds the performance of models 10 times larger on certain metrics. All the code and `Conifer` dataset are available at `https://www.github.com/ConiferLM/Conifer`.

## 1 Introduction

Large language models (LLMs) have achieved impressive performance across a wide range of natural language processing (NLP) tasks. Instruction tuning, also known as supervised fine-tuning (SFT) (Ouyang et al., 2022), has enabled LLMs to better align with human preferences by following human instructions and generating helpful, honest, and harmless (Askell et al., 2021) responses.

Improving instruction-following abilities is a core direction of concern for the LLM community. Recent progress in the field has enabled open-source LLMs to demonstrate impressive performance in following instructions across numerous tasks (Li et al., 2023; Zheng et al., 2023). Nonetheless, LLMs, particularly those that are open source, still often struggle with more challenging tasks that include complex constraints in the instructions (Sun et al., 2023; Jiang et al., 2023b; Qin et al., 2024). Figure 1 presents an instance illustrating an open source LLM's failure to adhere to instructions with multiple constraints. However, the challenge of enhancing LLMs to follow complex constraints is an area that remains insufficiently explored.

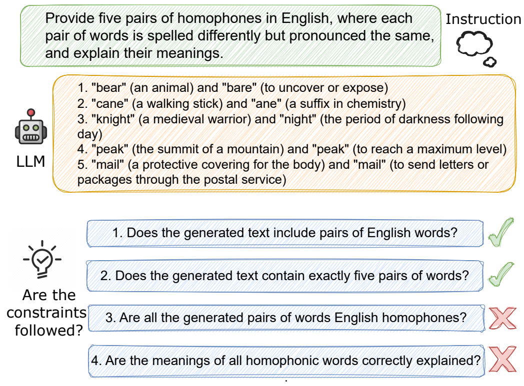



Figure 1: An example from InFoBench of how an LLM fails to follow the complex constraints in the instruction.

Recent studies have highlighted the importance of diverse and high-quality data in the success of instruction tuning (Zhou et al., 2023a; Liu et al., 2024). Instead of relying on the costly process of manual data annotation, researchers are utilizing high-performing models like ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) to generate instruction tuning datasets (Taori et al., 2023; Chiang et al., 2023; Peng et al., 2023; Xu et al., 2024; Ding et al., 2023a). This approach demonstrates performance comparable to manually annotated high-quality data. Nonetheless, there is a lack of research in the area focusing on enhancing LLMs' capacity to handle complex constrained instructions. It is also challenging to prompt GPT-4 directly to

*Equal contribution

generate instructions with complex constraints.

In this paper, we introduce a novel method for automatically generating an instruction-following dataset with complex constraints and construct the dataset called 🌲 **Conifer** (Complex **Con**strained **I**nstruction **F**ollowing). This dataset is constructed using the capabilities of GPT-4 with a random selection of user queries as seeds from ShareGPT. To address the challenge of generating instructions with multiple complex constraints, we break the hard task into smaller, more manageable tasks. These include query reframing, constraint generation, recombination, and two-stage filtering processes. Additionally, to facilitate the instruction-tuned models to effectively learn the difficult tasks from data, we propose a progressive learning scheme. By organizing the data into a multi-turn conversational format following an ascending difficulty level, the model effectively learn through an easy-to-hard progression Moreover, the model is enabled to learn from both internal and external process feedback, utilizing GPT-4's insights on the explicit reasoning process required to follow complex constraints.

The effectiveness of Conifer is validated by applying instruction tuning to the Mistral (Jiang et al., 2023a) and LLaMA-2 (Touvron et al., 2023) models, employing a combined dataset that merges 53k ShareGPT with 13k Conifer data, and further train the models with direct preference optimization (DPO) (Rafailov et al., 2023). The efficacy of our methodology is assessed on the recently proposed instruction-following benchmarks, including IFEval (Zhou et al., 2023b), FollowBench (Jiang et al., 2023b) and InFoBench (Qin et al., 2024), where instructions are notably more complex and come with constraints. Additionally, it was benchmarked on widely-recognized AlpacaEval (Li et al., 2023) and MT-Bench (Zheng et al., 2023). The models trained with our Conifer dataset showed significant improvements in following complex and constrained instructions compared to its 7B counterparts. Notably, the performance of Conifer-7B-DPO even outperforms the best open-source models at the 70B scale on IFEval and FollowBench Hard. The main contributions of the paper are summarized as follows:

- We tackle a critical yet under-explored challenge for LLMs: their difficulty in complying with complex, constrained instructions.
- We introduce a new paradigm to generate the instructions with complex constraints, and release the Conifer, a novel instruction tuning dataset, specifically designed to enhance LLMs' capabilities in following complex constrained instructions. To construct this dataset, we decompose the complex task into smaller, more manageable tasks for GPT-4 to execute. We also introduce a progressive learning method that helps models to develop the ability to interpret instructions, by an easy-to-hard progression and learning from explicit internal and external process feedback.
- Extensive experiments, including ablation studies, demonstrate the effectiveness of the proposed approach. LLMs trained with the Conifer dataset achieve impressive performance in instruction-following ability, especially especially when dealing with complex and constrained instructions, as evidenced by various instruction-following benchmarks.

## 2 Related Work

**Instruction Tuning** Instruction tuning, or supervised fine-tuning (SFT) is to fine-tune LLMs to follow users' instructions, making them more controllable and predictable (Zhang et al., 2023b). The SFT model can be further trained to better align with humans using reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022) or direct preference optimization (DPO) (Rafailov et al., 2023).

Conventional approaches (Mishra et al., 2022; Wei et al., 2021; Sanh et al., 2021; Lou et al., 2024) aggregate data from a large amount of NLP tasks to train multi-task models and that achieve strong results in downstream tasks. However, studies have highlighted a notable discrepancy between these NLP tasks and the actual way users interact with LLMs (Ouyang et al., 2023), leading to responses less preferred by humans (Zheng et al., 2023; Li et al., 2023). In contrast, recent researches have been directed towards constructing instruction tuning datasets that better reflect humans' intent. These datasets can be either manually annotated (Ouyang et al., 2022; Kopf et al., 2023; Zhou et al., 2023a) or generated by advanced LLMs like GPT-4 (Peng et al., 2023). The generation of synthetic data through LLMs has emerged as a cost-effective research approach, highlighted by studies such as Alpaca (Taori et al., 2023), which utilizes self-instruct (Wang et al., 2022) to prompt ChatGPT from a

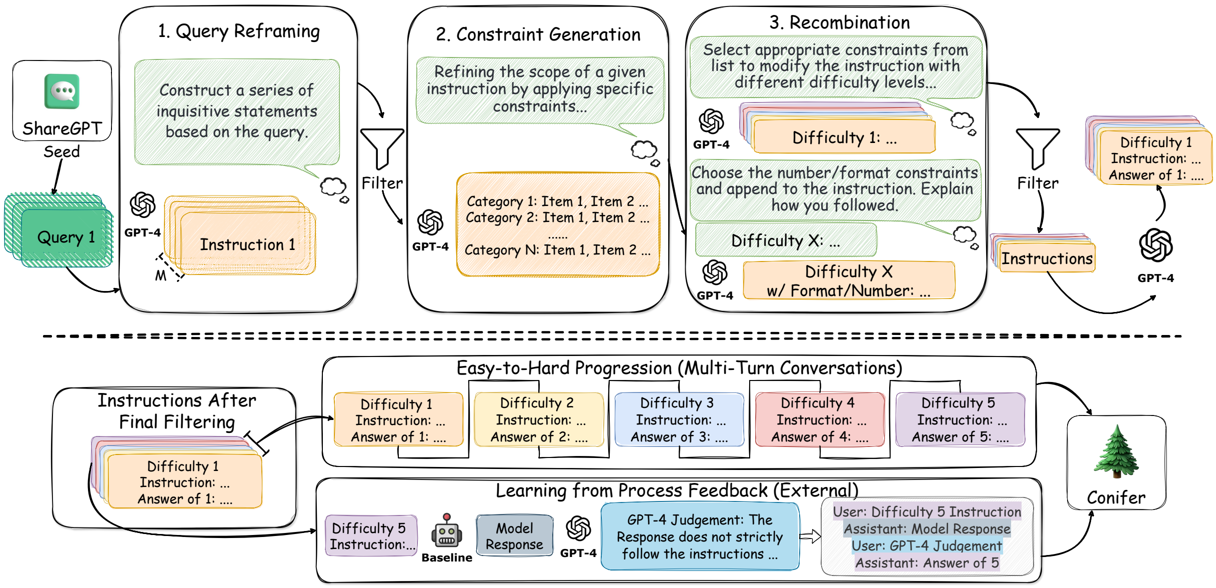



Figure 2: Paradigm of the production of the proposed Conifer dataset. The upper portion depicts the instruction collection phase (Section 3.1), while the lower portion outlines the progressive learning scheme (Section 3.2).

range of seed tasks. Vicuna (Chiang et al., 2023) and other studies (Geng et al., 2023; Wang et al., 2023) investigate the use of ShareGPT data, which more closely mirrors human intentions (Ouyang et al., 2023) and has demonstrated increased efficacy (Ivison et al., 2023). Additional methods leverage LLMs to simulate conversations (Ding et al., 2023b) or to undertake instruction expansion (Lou et al., 2024) for generating SFT data.

**Instruction Following with Complex Instructions or Constraints** Traditional approaches for controllable text generation (Zhang et al., 2023a) often involve fine-tuning LMs on specific tasks (Zhang et al., 2020) or design post-processing strategies (Dathathri et al., 2020). In the era of LLM, enhancing controllable text generation is often achieved by improving the instruction-following capabilities of LLMs. Several recent studies investigate the ability of LLMs for instruction following, presenting new benchmarks (Jiang et al., 2023b; Qin et al., 2024; Zhou et al., 2023b). These studies conclude that current LLMs struggle at meeting fine-grained or hard constraints (Sun et al., 2023). Despite the significance of this challenge, few studies aim explicitly at addressing it. Most related, WizardLM (Xu et al., 2024) aims to enhance the overall ability of LLMs with different task complexity. The proposed Evol-Instruct is created by in-breadth and in-depth transformation, which may alter the semantic of original instructions. But our paper mainly focuses on improving LLMs to follow instructions with complex constraints, achieved by adding constraints and preserving the core entities unchanged, so the scope of constraints here is substantially broader, more challenging, and more numerous than those in WizardLM.

## 3 The Conifer Dataset

In this section, we detail the methodologies employed in creating the Conifer dataset. We illustrate the complete data construction pipeline in Figure 2, consisting of two principal stages: instruction collection from GPT-4 (Section 3.1) and organizing the dataset into a progressive learning scheme (Section 3.2). Throughout all stages involving the LLM, we utilize GPT-4 Turbo to execute the tasks.

### 3.1 Instruction Collection

Building on the successful efforts of previous work (Taori et al., 2023; Xu et al., 2024; Lou et al., 2024) and the analysis in Section 1, our goal is to generate instructions that are not only high in quality but also rich in diversity, complexity, and constraints. Therefore, we have randomly selected 6,000 user queries (prompts) from ShareGPT as our seed instructions. Given that ShareGPT data is sourced from open-domain conversations with ChatGPT, it offers a wide variety of topics, ensuring sufficient diversity of the Conifer dataset. Our preliminary experiments indicate that GPT-4 consistently struggles in generating instructions that contain multiple complex constraints. To address this challenge and to enhance the diversity and complexity of the Conifer dataset, we have decomposed this challenging task into smaller, more manageable tasks for GPT-4, including query reframing, constraint generation, recombination, and a two-stage filtering process.

**Query Reframing** Our primary objective is to construct a dataset rich in complex constraints, targeting a diverse array of constraint types. To realize this, we initiate a process of query reframing to diversify the seed instructions. This involves editing the instructions to provide alternative perspectives while keeping the core entities unchanged. We utilize GPT-4 to reformulate each query into at least three distinct and varied forms, thereby enriching the dataset with a broader range of perspectives.

**Constraint Generation** After the query reframing phase, we direct GPT-4 to identify the subjects and objects within the rephrased instructions and to generate a list of potential constraints to narrow down response options. Our preliminary tests have indicated that GPT-4 struggles to produce appropriate constraints consistently. Consequently, we have adopted in-context learning techniques, providing manually crafted examples that pair instructions with their corresponding constraints. These constraints are categorized into two levels: broader categories and specific items. For instance, in naming-related instructions, 'Cultural Background' could serve as a category, with 'Chinese', 'Continental' as example items within this category. Organizing constraints in this way greatly assists us in prompting GPT-4 more effectively in the subsequent stages.

**Recombination** In this stage, for each instruction derived from the query reframing phase, GPT-4 is directed to select specific and suitable constraints from the generated constraint list and incorporate them into the instruction. This systematic modification facilitates the creation of instructions bound by predetermined constraints.

To increase the complexity of the instructions, we instruct GPT-4 to develop instructions with varying levels of difficulty. The level of difficulty is quantified by the number of constraints incorporated into each instruction. Each instruction of increasing difficulty should integrate 1-2 categories and 2-3 items, thus presenting a more complex challenge than its simpler counterpart. The scale of difficulty culminates at degree 5, indicating that each instruction derived from the query reframing stage will be crafted into up to five distinct instructions, reflecting a progressive scale of complexity.

Through observation, we notice that GPT-4 seems to struggle with incorporating format or numerical constraints with sentences while displaying a preference for adding content-based constraints. Since format and numerical constraints are important for instruction following, we randomly select 1,000 instructions after the recombination and prompt GPT-4 to augment them with additional format or numerical constraints. These constraints are synthesized by GPT-4 using three seed constraints. We curate multiple formatting and numerical constraints with the help of GPT-4. We manually review these synthesized constraints to ensure they are distinct from those in our evaluation benchmarks, after discarding any that are impractical or redundant, we obtained 32 constraints.

**Two-Stage Filtering** To ensure the quality of the generated instructions, we incorporate a two-stage filtering process utilizing GPT-4. The initial filter occurs subsequent to the query reframing, primarily targeting the removal of instructions that lack necessary context for meaningful responses. The second filter takes place after the recombination, concentrating on identifying and resolving conflicts within instructions that may result from the recombination of various constraints.

By applying the outlined production process, we collect 35,613 instructions from GPT-4, and further engaged the model to generate corresponding responses for each instruction.

### 3.2 Progressive Learning Scheme

Generating high-quality data with instructions and responses doesn't fully solve the problem of following complex instructions for LLMs. Given the dataset's high complexity, how to enable open-source, often smaller LLMs to learn how GPT-4 interpret complex instructions and tackle complex problems poses significant challenges. This is crucial ensure that these models don't merely imitating the style of proprietary language models (**?**) but also their accuracy and depth of understanding. To tackle this challenge, we have developed a progressive learning strategy. This strategy uses two main techniques for structuring the data: an easy-to-hard progression and learning from process feedback.

**Easy-to-Hard Progression** Inspired by curriculum learning (Bengio et al., 2009), which suggests the benefits of starting with simpler examples and gradually moving to more complex ones, we have accordingly organized the Conifer dataset. Specifically, we aggregate multi-level instructions and answers generated from the same seed instruction into a single sample using the multi-turn conversational format. The sequence is organized to introduce the simpler tasks first, which then progressively lead towards more challenging ones, as depicted in Figure 2.

**Learning from Process Feedback** Preliminary experiments reveal that fine-tuned models struggle with adhering to the most challenging samples, particularly with respect to specific format and numerical constraints. Inspired by the recent research on process supervision (Lightman et al., 2023), which exposes the supervision signal at each reasoning steps and supervise the model's thought step, We facilitate the model to learn through process feedback from both internal and external perspectives.

Internally, when prompted to respond to instructions that contain format and numerical constraints, GPT-4 is instructed to illustrate how it adheres to the specified constraints explicitly within its responses. The instructions of this part are randomly sampled from all the instructions of our Conifer.

Externally, GPT-4 is tasked to identify constraints where it fails to adhere in the most challenging constraints, and this judgement is employed to construct a new multi-turn conversation, which take the form of: {Difficult Instruction, Model Response, GPT-4 Judgement, GPT-4 Response}. The instructions of this part are selectively sampled from the highest difficulty level, level 5, within the Conifer dataset.

By incorporating internal and external feedback by GPT-4's evaluations, models gain additional supervision signal on reasoning processes, enhancing their ability to interpret and follow complex constraints.

### 3.3 Statistics of the Conifer Dataset

Table 1 presents the Conifer dataset statistics, including the counts for the easy-to-hard progression ('Multi-Turn') and the learning from process feedback ('Feedback') parts. The 'Multi-Turn' part includes instructions with different degrees of difficulty (DD) ranging from 1 to 5 for the easy-to-hard progression, noting that not all multi-turn conversations contain all five levels. Additionally, the table lists learning from process feedback as internal and external feedback. There are totally 13,606 conversations in the Conifer dataset with an 3.02 average number of turns per conversation.

| Class | # Samples | Subclass | # Ins. |
|---|---|---|---|
| Easy-to-Hard | 10302 | DD 1 | 9167 |
| | | DD 2 | 8146 |
| | | DD 3 | 7157 |
| | | DD 4 | 6144 |
| | | DD 5 | 5017 |
| Process Feedback | 3304 | Internal | 977 |
| | | External | 2327 |

Table 1: Statistics of the Conifer dataset. # Ins. denotes the number of instructions.

We also evaluate our Conifer's complexity and quality with the Deita complexity/quality scorer (Liu et al., 2024). Relative to ShareGPT, Conifer demonstrates marked improvements in both aspects, as detailed in Figure 5.

Further analysis of the Conifer dataset, including examples of filtering cases, is shown in Appendix A. The Conifer dataset has been released to the public on HuggingFace `https://huggingface.co/datasets/ConiferLM/Conifer`.

## 4 Experiments

Comprehensive experiments are conducted to evaluate the performance of our proposed method and Conifer dataset, with a particular focus on improving the ability of models to follow complex constrained instructions.

### 4.1 Models and Datasets

We conduct experiments on two popular base LLMs, **Mistral-7B** (Jiang et al., 2023a) and **LLaMA-2-13B** (Touvron et al., 2023). Mistral-7B is the state-of-the-art base large language model at the 7B parameter scale.

Since the Conifer dataset mainly address the instruction-following ability of LLMs under complex constraints, we merge the collected 13k Conifer dataset with the initial 53k ShareGPT[1] to form the final 66k SFT dataset, similar to the strategy of WizardLM (Xu et al., 2024).

Although our primary focus is on the SFT alignment stage using the proposed Conifer dataset, we

[1]Collected from `https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered` and after language filtering, we obtain 53k samples.

| Model | Base Model | Final Stage | IFEval | FollowBench (HSR) | | | | | | InFoBench | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | loose prompt | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Avg | Easy | Hard | Overall |
| GPT-4† | - | - | 79.3 | 84.7 | 76.1 | 71.3 | 74.5 | 62.4 | 73.8 | 90.1 | 89.1 | 89.4 |
| GPT-3.5 Turbo† | - | - | - | 80.3 | 68.0 | 68.6 | 61.1 | 53.2 | 66.2 | 90.4 | 85.1 | 86.7 |
| Qwen-72B-Chat† | Qwen | - | 50.8 | 73.8 | 63.3 | 54.3 | 45.2 | 39.9 | 55.3 | - | - | - |
| LLaMA-2-70B-Chat† | LLaMA-2 | RLHF | - | 59.9 | 53.3 | 46.0 | 40.2 | 37.9 | 47.5 | 89.6 | 82.1 | 84.4 |
| Vicuna-13B-v1.5 | LLaMA-2 | SFT | 46.6 | 71.2 | 61.3 | 48.3 | 38.0 | 33.1 | 50.4 | 85.7 | 73.7 | 77.3 |
| LLaMA-2-13B-ShareGPT | LLaMA-2 | SFT | 47.1 | 59.2 | 48.7 | 45.8 | 30.9 | 27.4 | 42.4 | 84.3 | 75.4 | 78.2 |
| Conifer-13B | LLaMA-2 | SFT | 47.5 | 60.5 | 53.6 | 48.4 | 40.7 | 31.7 | 47.0 | 84.5 | 76.5 | 78.9 |
| Deita-7B-V1.0-SFT | Mistral | SFT | 45.1 | 55.8 | 51.3 | 39.9 | 32.6 | 30.6 | 42.0 | 84.8 | 75.9 | 78.6 |
| Zephyr-7B-beta | Mistral | DPO | 44.9 | 57.6 | 51.9 | 41.9 | 41.4 | 31.4 | 44.8 | 84.1 | 75.3 | 78.0 |
| Deita-7B-V1.0 | Mistral | DPO | 51.9 | 55.8 | 49.3 | 46.7 | 39.6 | 37.3 | 45.7 | 86.2 | 78.6 | 80.9 |
| Mistral-7B-Muffin | Mistral | SFT | 34.0 | 40.1 | 31.5 | 23.9 | 17.8 | 14.4 | 25.6 | 70.0 | 66.9 | 67.8 |
| Mistral-7B-Evol-Instruct | Mistral | SFT | 44.0 | 53.2 | 51.0 | 44.3 | 31.7 | 23.5 | 40.7 | 81.0 | 73.2 | 75.6 |
| Mistral-7B-ShareGPT | Mistral | SFT | 43.4 | 55.7 | 51.2 | 43.0 | 38.2 | 26.4 | 42.9 | 84.5 | 75.8 | 78.5 |
| Conifer-7B | Mistral | SFT | 50.8 | 54.3 | 49.5 | **49.3** | 40.8 | 30.5 | 44.9 | 83.6 | 77.7 | 79.5 |
| Mistral-7B-ShareGPT-DPO | Mistral | DPO | 48.2 | 58.4 | **53.9** | 48.3 | 39.1 | 38.6 | 47.7 | 86.8 | 79.9 | 82.0 |
| Conifer-7B-DPO | Mistral | DPO | **52.3** | **60.3** | 53.6 | 48.0 | **47.1** | **41.0** | **50.0** | **87.5** | **80.0** | **82.3** |

Table 2: Main results on three instruction-following benchmarks: IFEval, FollowBench, and InFoBench. We use boldface for the best results and underline for the second-best results among the 7B models. † indicates that the results are directly sourced from the original benchmarks.

also explore the benefits of direct preference optimization (DPO) (Rafailov et al., 2023) training, which has shown significant improvements. Following previous work (Tunstall et al., 2023b; Liu et al., 2024), we utilize the UltraFeedback dataset (Cui et al., 2023) for DPO training.

### 4.2 Evaluation

We focused our evaluation on three challenging instruction-following benchmarks. Comparing with other benchmarks, instructions in these benchmarks are typically harder and often contains constraints.

- **IFEval** (Zhou et al., 2023b) evaluates instruction-following LLMs with various verifiable instructions. Instructions from IFEval contain many lexical and format constraints.
- **FollowBench** (Jiang et al., 2023b) is a multi-level fine-grained constraints following benchmark, including five different types (Content, Situation, Style, Format, and Example) and five difficulty levels, with performance assessed using GPT-4.
- **InFoBench** (Qin et al., 2024) is a benchmark that breaks down a complex constraint into simpler criteria across multiple categories. It also leverages GPT-4 for evaluation process.

We also conduct evaluations using two widely recognized instruction-following benchmarks on LLMs' general ability to align with human preferences.

- **AlpacaEval** (Li et al., 2023) is an LLM-based automatic evaluation by GPT-4 Turbo. The score is the win-rate against the reference model (GPT-4 Turbo) on the dataset. We adopt AlpacaEval 2.0 Length-Adjusted (LC) win rate (Dubois et al., 2024), the latest version of AlpacaEval, which has a spearman correlation of 0.98 with ChatBot Arena.
- **MT-Bench** (Zheng et al., 2023) is multi-turn question set containing 80 multi-turn questions. The model answers an initial question, and then responds to a predefined another question. The responses are rated by GPT-4 on scores scaled from 1-10. We follow the guidelines from the authors to use GPT-4 0314 version for judgement.

Alignment with human preference is the main objective for instruction tuning. The two benchmarks are effective measures for evaluating how well models align with general user preferences, showing high correlation with human judgments.

We compare our Conifer models with various open-source LLMs. Specifically, we choose two 70B size models: Qwen-72B-Chat (Bai et al., 2023) and LLaMA-2-70B-Chat (Touvron et al., 2023) which are among the best open-source LLMs. All the compared open-source 7B LLMs have released their instruction tuning data, including Deita (Liu et al., 2024) and Zephyr-7B-beta (Tunstall et al., 2023b) models which establish SOTA results through DPO alignment based on Mistral-7B. In 13B experiments, we compare with Vicuna-13B-

| Model | Base Model | Stage | AlpacaEval 2.0 LC Win Rate | AlpacaEval 2.0 Avg Length | MT-Bench Score |
|---|---|---|---|---|---|
| GPT-4 0613† | - | | 30.2% | 1140 | 9.18 |
| GPT-3.5 Turbo 0613† | - | - | 22.7% | 1328 | 8.39 |
| Deita-7B-v1.0-SFT† | Mistral | SFT | - | - | 7.22 |
| Deita-7B-v1.0† | Mistral | DPO | 16.1% | 1417 | **7.55** |
| Zephyr-7B-beta† | Mistral | DPO | 13.2% | 1444 | 7.34 |
| Mistral-7B-Muffin | Mistral | SFT | 3.8% | 736 | 3.90 |
| Mistral-7B-Evol-Instruct | Mistral | SFT | 9.4% | 982 | 6.51 |
| Mistral-7B-ShareGPT | Mistral | SFT | 11.6% | 1070 | 6.86 |
| Conifer-7B | Mistral | SFT | 12.5% | 1052 | 7.08 |
| Mistral-7B-ShareGPT-DPO | Mistral | DPO | 15.1% | 1276 | 7.10 |
| Conifer-7B-DPO | Mistral | DPO | **17.1%** | 1253 | 7.25 |

Table 3: Evaluation on the AlpacaEval and MT-Bench benchmarks for general LLM instruction-following ability. † indicates that the scores are directly sourced from the original benchmarks.

v1.5 (Chiang et al., 2023) and our trained LLaMA-2-13B on 53k ShareGPT data only, since Vicuna's 125k ShareGPT dataset has not been released yet.

Conifer is also compared with two researches that are most related to ours: WizardLM (Xu et al., 2024), and Muffin (Lou et al., 2024). We train baseline models using only the ShareGPT dataset, the Muffin dataset, and as well as combining ShareGPT with Evol-Instruct (following the WizardLM paper), keeping the base LLM and experimental settings consistent with ours for fair comparisons.

Other experimental details, including (1) details of training, (2) details of baselines, (3) full results on the IFEval and FollowBench, and (4) additional results on the Open LLM leaderboard are shown in Appendix B.

### 4.3 Main Results

Table 2 presents our results on three instruction-following benchmarks: IFEval, FollowBench, and InFoBench. The Conifer-7B-DPO model exhibits exceptional performance across all benchmarks, even outperforming or matching the capabilities of 70B models. Specifically, it achieves state-of-the-art performance on the IFEval benchmark, and achieves an excellent average score on FollowBench, surpassing LLaMA-2-70B-Chat and trailing just behind Qwen-72B-Chat. Our model's proficiency is particularly evident in handling difficult constraints, with its excellent performance in the Level 4 to Level 5 constraint difficulty on FollowBench. Specifically, Conifer-7B-DPO obtains a 41.0% success rate on Level 5, outperforming Qwen-72B-Chat's 39.9%. On InFoBench, it continues its remarkable performance against all open-source models of similar size, and close the gap between 7B models and LLaMA-2-70B-Chat, confirming Conifer's superiority in handling challenging tasks.

Within the 7B model category, our primary comparisons involve state-of-the-art Mistral-7B models like Zephyr and Deita, as well as WizardLM (Evol-Instruct) and Muffin. Comparisons reveal that Conifer-7B outperforms all the aforementioned models in a substantial margin. Compared to Deita-7B and Mistral-7B-Evol-Instruct, which are trained on datasets noted for their high quality and complexity, our Conifer-7B showcases remarkable superiority, affirming the exceptional performance of the Conifer dataset in improving the ability of LLMs in follow constrained instruction-following tasks.

Comparing to models trained with ShareGPT dataset only, Conifer surpasses them across all benchmarks with the same 7B or 13B base LLM, under both the SFT and DPO stages. Note that models we trained on LLaMA-2-13B do not surpass Vicuna-13B-v1.5 on FollowBench, which may be attributed to Vicuna's use of more ShareGPT data (125k) that has not been made public.

In Table 3, we also illustrate the performance on AlpacaEval and MT-Bench benchmarks that mainly assess LLMs' overall ability to align with human preferences. From this table, our Conifer model outperforms baseline models at the same stage, such as ShareGPT, Evol-Instruct, and Muffin, indicating consistent improvements in aligning with general human preferences.

While AlpacaEval and MT-Bench benchmarks may correlate with response lengths (Zhao et al., 2024), Conifer models do not generate lengthy responses. The average length is much shorter than

| Model | IFEval | FollowBench | | |
|---|---|---|---|---|
| | | L1-L3 | L4-L5 | Avg |
| w/o Conifer | 43.4 | 50.0 | 32.3 | 42.9 |
| random shuffle | 48.6 | 48.1 | 38.2 | 44.1 |
| hard-to-easy | 49.9 | 47.5 | 33.1 | 41.7 |
| easy only | 47.5 | 51.3 | 33.5 | 44.2 |
| hard only | 48.2 | 49.5 | 34.3 | 43.4 |
| single turn | 46.2 | 51.5 | 33.1 | 44.2 |
| w/o process feedback | 44.2 | 48.4 | 36.1 | 43.5 |
| w/ internal only | 49.4 | 48.0 | 37.2 | 43.7 |
| w/ external only | 48.2 | 51.6 | 35.7 | 45.3 |
| w/o format&number | 48.6 | 51.4 | 35.2 | 44.9 |
| Conifer | 50.8 | 51.0 | 35.6 | 44.9 |

Table 4: Ablation studies results when applying different progressive learning scheme in Section 3.2.

those produced by Deita and Zephyr models. The latest AlpacaEval LC win rate solves the length bias issue. Conifer-7B-DPO achieves a high 17.1% LC Win Rate, outperforming other open-source 7B models like Zephyr-7B-beta and Deita-7B-v1.0. These validate Conifer's competitive performance within the scope of open-source 7B models.

### 4.4 Ablation Studies

Ablation studies are carried out to investigate the impact of the progressive learning scheme introduced in Section 3.2, with Mistral-7B as base LLM and applying SFT for alignment.

The results, summarized in Table 4, reveal that varying the sequence of difficulty within the easy-to-hard progression, such as through random shuffling or reversing the order to hard-to-easy, results in reduced overall performance. In particular, randomizing the sequence results in a notable decrease in performance. Excluding certain difficulty levels, such as including only easy or only hard instructions, leads to decreased performance on the L4-L5 or L1-L3 of FollowBench, respectively. Removing the easy-to-hard progression learning scheme, by using a single turn instead of a multi-turn format, diminishes the model's effectiveness on instruction following capability.

The significance of learning from process feedback is also completely examined. It is clear that each component plays a crucial role, as demonstrated in Table 4. Note that relying solely on internal feedback results in a notable decrease in FollowBench, while the IFEval score slightly improves, which is in line with expectations since internal feedback only addresses format and numerical constraints where IFEval is particularly

| Train Set | Test Set | Rephrased Samples | Percentage (%) |
|---|---|---|---|
| Conifer | IFEval | 3 | 0.55 |
| Conifer | FollowBench | 5 | 0.53 |
| Conifer | InFoBench | 11 | 2.20 |
| ShareGPT | IFEval | 29 | 5.36 |
| ShareGPT | FollowBench | 59 | 6.25 |
| ShareGPT | InFoBench | 36 | 7.20 |

Table 5: Rephrased samples assessment between Conifer and ShareGPT datasets with the benchmarks, as evaluated by GPT-4 Turbo. The values indicate the proportion of rephrased samples within each test set.

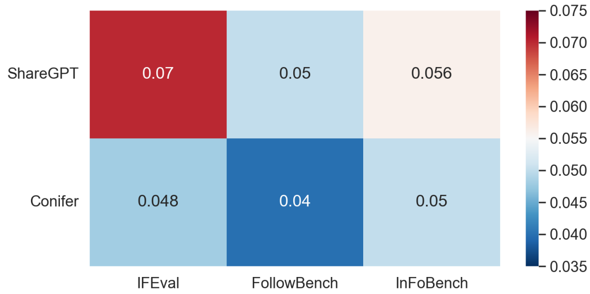



Figure 3: Cosine similarities between sentences from training and testing sets. Values near 0 indicate little overlap and no data contamination.

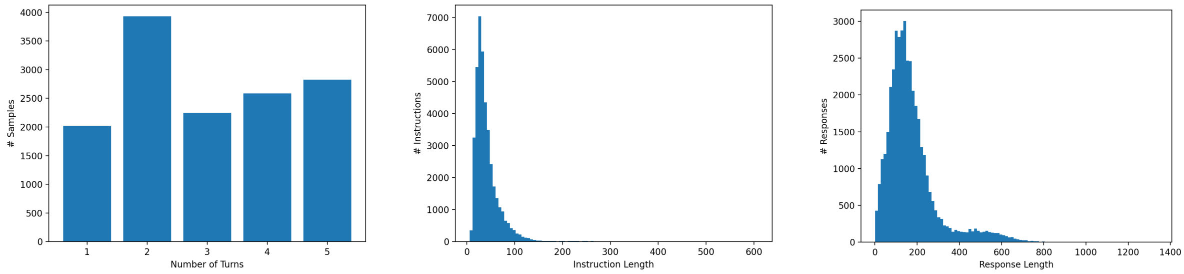
focused. Similarly, the slight performance boost at the L4-L5 level of FollowBench when only external feedback is provided can align with the understanding that such feedback is most effective for the hardest instructions. Incorporating both internal and external process feedback is able to achieve comprehensive performance improvements across IFEval and FollowBench.

### 4.5 Is There Any Data Contamination?

Data contamination has been a critical problem in the era of LLMs, where training on the test set will definitely bring improvement (Wei et al., 2023; Deng et al., 2023). To assess whether Conifer dataset exhibits data contamination with three instruction-following benchmarks, we conduct analysis using: (1) cosine similarity between Conifer and test samples (Lou et al., 2024); (2) using GPT-4 to detect rephrased samples across training and testing sets (Yang et al., 2023).

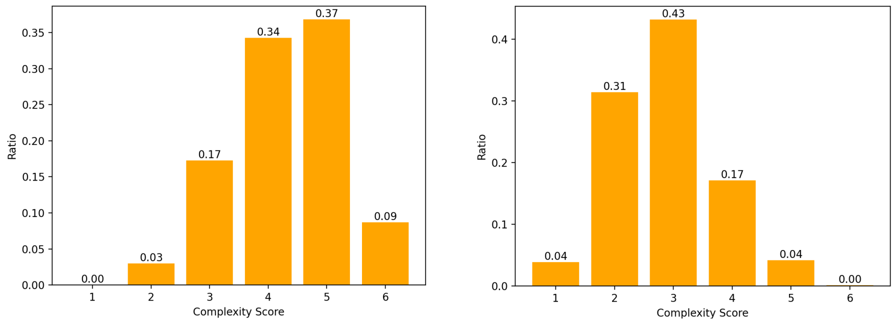
Figure 3 illustrates sentence-level cosine similarities of embeddings, which is calculated by Sentence Transformers[2]. We compare the Conifer against ShareGPT across the IFEval, FollowBench,

[2]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

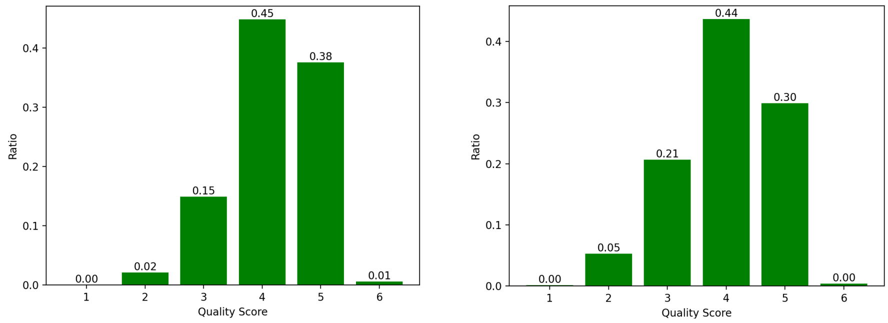



(a) Distribution of Turn (b) Distribution of Instruction Length (c) Distribution of Response Length

Figure 4: Distribution of turns round, instructions length, and responses length in Conifer dataset.




(a) Conifer Complexity Score (b) ShareGPT Complexity Score




(c) Conifer Quality Score (d) ShareGPT Quality Score

Figure 5: Distribution of complexity/quality score of our Conifer dataset and ShareGPT data, which is measured by Deita complexity/quality scorer.

and InFoBench test sets and get lower similarity scores, suggesting minimal data overlap.

Additionally, following recent work on data contamination detection (Yang et al., 2023), we employ GPT-4 to identify rephrased samples from Conifer dataset and the benchmarks, results are in Table 5. Compared to ShareGPT dataset, Conifer shows relatively lower percentage of similar samples, which indicates an absence of data contamination.

### 4.6 Quantity, Quality, and Complexity

To give a better illustration of the quantity of our Conifer dataset, the distribution of the number of turns, instruction length, and response length in our Conifer, is shown in Figure 4. All the length is counted by words using NLTK toolkit.

To examine the quality and complexity of our Conifer, we utilize the Deita quality/complexity scorer (Liu et al., 2024) which is trained based on LLaMA. The distribution of quality and complexity are shown in Figure 5. It should be noted that the scorer might have bias. We also calculate both scores of ShareGPT data. From the comparison, we can conclude that Conifer exhibit more complexity and higher quality than ShareGPT.

## 5 Conclusion

In this paper, we tackle the important yet under-explored challenge for LLMs: difficulty in following complex, constrained instructions. We create a novel instruction tuning dataset, Conifer, with GPT-4's assistance, and carefully curate it to ensure quality. We propose an effective progressive learning scheme, with easy-to-hard progression and learning from progress feedback. Experimental results demonstrate that Conifer-7B-DPO outperforms the best open-source 7B models according to instruction-following benchmarks, even matches the performance of models 10 times larger on certain metrics..

## References


2019. Winogrande: An adversarial winograd schema challenge at scale.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, and others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. Published: Blog post.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*.

Ning Ding, Yulin Chen, Bokai Xu, Shengding Hu, Yujia Qin, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023a. UltraChat: A Large-scale Auto-generated Multi-round Dialogue Data. Published: GitHub Repository.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023b. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. *arXiv preprint arXiv:2305.14233*.

Yann Dubois, Balazs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-corrected alpacaeval: A simple debiasing of automatic evaluators. `https://github.com/tatsu-lab/alpaca_eval`.

Tianyu Gao. 2023. Teach llamas to talk: Recent progress in instruction tuning.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning AI With Shared Human Values. In *ICLR*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

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

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2023b. FollowBench: A Multi-level Fine-grained Constraints Following Benchmark for Large Language Models. *arXiv preprint arXiv:2310.20410*.

Andreas Kopf, Yannic Kilcher, Dimitri von Rutte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Rich'ard Nagyfi, E. S. Shahul, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. OpenAssistant Conversations - Democratizing Large Language Model Alignment. *ArXiv*, abs/2304.07327.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. Publication Title: GitHub repository.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and others. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu su, and Wenpeng Yin. 2024. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774. ArXiv: 2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Siru Ouyang, Shuohang Wang, Yang Liu, Ming Zhong, Yizhu Jiao, Dan Iter, Reid Pryzant, Chenguang Zhu, Heng Ji, and Jiawei Han. 2023. The shifted and the overlooked: A task-oriented investigation of user-GPT interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2375–2393, Singapore. Association for Computational Linguistics.

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

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating Large Language Models on Controlled Generation Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

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. Published: GitHub repository.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. 2023a. The alignment handbook. `https://github.com/huggingface/alignment-handbook`.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, and others. 2023b. Zephyr: Direct Distillation of LM Alignment. *arXiv preprint arXiv:2310.16944*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. *arXiv preprint arXiv:2309.11235*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning Language Model with Self Generated Instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. 2023. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2024. Wizardlm: Empowering large language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *The Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023a. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

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

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and William B Dolan. 2020. Pointer: Constrained Text Generation via Insertion-based Generative Pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8649–8670.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *NeurIPS Datasets and Benchmarks Track*. _eprint: 2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and others. 2023a. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

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. *arXiv preprint arXiv:2311.07911*.

# A Details of Conifer Dataset

## A.1 Dataset Filtering Cases

Table 6 demonstrates cases filtered through the two-stage filtering process, and it includes explanations from GPT-4 Turbo on why certain cases were removed.

# B Details of Experiments

## B.1 Details of Training

For model training, we utilize the widely-used repository 'The Alignment Handbook' (Tunstall et al., 2023a), released by HuggingFaceH4, to train Mistral-7B based models, and use FastChat (Zheng et al., 2023) to train LLaMA-2-13B based models. The Mistral-7B based experiments in the paper are done using 8 NVIDIA A100 80GB GPUs. We adopt DeepSpeed ZeRO stage 2 for SFT and DPO training. The SFT models are trained with 4 epochs. We set the learning rate to $2.0e{-}05$, with per device batch size 16 and gradient accumulation steps 4. The warm-up ratio is set to 0.1. While LLaMA-2-13B based experiments are done using 16 NVIDIA A100 80GB GPUs, trained with 3 epochs, per device batch size 2, gradient accumulation steps 4, and warm-up ratio is set to 0.03. For DPO training, the beta is set to 0.01 and trained with 1 epoch. The learning rate is set to $5.0e-07$, with per device batch size 4 and gradient accumulation steps 4. We apply cosine learning rate scheduler and gradient checkpointing in the experiments and the max sequence length is set to 2048.

## B.2 Details of Baselines

The baseline methods in the experiments include:

- **Qwen-72B-Chat** (Bai et al., 2023), developed by Alibaba Cloud, is among the best open-source LLMs, which is pretrained on 3T tokens. However, the dataset adopted for alignment of the model is undisclosed.

- **LLaMA-2-70B-Chat** (Touvron et al., 2023) is a powerful open-source model released by Meta. It a fine-tuned version of LLaMA-2-70B that is optimized using SFT and RLHF for dialogue use cases. The datasets for training this model is closed-source.

- **Vicuna-13B-v1.5** (Chiang et al., 2023) is a powerful LLaMA-2-13B based model which trained on 125k ShareGPT data, but the training data will not release.

- **Zephyr-7B-beta** (Tunstall et al., 2023b) is a state-of-the-art LLM based on Mistral-7B. Zephyr-7B-beta has been fine-tuned on UltraChat (Ding et al., 2023b) instruction tuning dataset and optimized with DPO on the UltraFeedback (Cui et al., 2023) dataset.

- **Deita-7B-1.0-SFT** is another state-of-the-art LLM based on Mistral-7B. Deita proposes a data selection strategy and curate a 6K high-quality subset from ShareGPT, UltraChat, and WizardLM Evol-Instruct (Xu et al., 2024) datasets. So the general SFT data is better than ShareGPT used in Conifer. However, Conifer can be further combined with data selection methods like Deita to achieve better overall performances. **Deita-7B-V1.0** (Liu et al., 2024) is further aligned using DPO on the UltraFeedback dataset.

- **LLaMA-2-13B-ShareGPT** and **Mistral-7B-ShareGPT** are models we train with the publicly available 53k ShareGPT instruction tuning dataset, using LLaMA-2-13B and Mistral-7B as base models respectively. We further apply DPO training to get **Mistral-7B-ShareGPT-DPO** under the same UltraFeedback dataset and experimental conditions as Conifer for fair comparisons.

- **Mistral-7B-Evol-Instruct** (Xu et al., 2024): We have trained this model on the publicly available WizardLM-Evol-Instruct dataset[3], with 143k sample containing mixture evolved data of Alpaca and ShareGPT. We follow the original paper (Xu et al., 2024) to combine 53k ShareGPT dataset with 143k evol-instruct dataset to get final 196k full dataset.

- **Mistral-7B-Muffin**: We trained the model using the Muffin dataset, with a combination with ShareGPT dataset to train Mistral-7B for fair comparison with Conifer. We also try only adopt Muffin dataset and get inferior results. Muffin is an instruction-following dataset aligning with scaling tasks per input. Multi-faceted instructions are curated with the assistance of ChatGPT and GPT-4.

[3] https://huggingface.co/datasets/WizardLM/WizardLM_evol_instruct_V2_196k

| | | |
|---|---|---|
| **First Filter** | | |
| Case 1 | **Query**: | How does the noise cancellation technology in the Bose QuietComfort 35 II compare to other models? |
| | **Filter Response**: | No, because providing a comparison **requires up-to-date** and specific information about other models' noise cancellation technology, which may have evolved since my last update, and without knowing which specific models you're referring to, a detailed comparison cannot be made. |
| Case 2 | **Query**: | What is the total amount of payroll if 75% is paid in checks? |
| | **Filter Response**: | No, because the total amount of payroll is not specified, and **without knowing the total or the actual amount** paid in checks, it is impossible to calculate the total payroll. |
| **Final Filter** | | |
| Case 1 | **Instruction**: | What technical manuals offer guidance on using paragraphs versus numbered steps for novice users? |
| | **Filter Response**: | No, because the question is asking for specific technical manuals, which could be numerous and varied across different fields and industries. **Without additional context** about the field or type of technology, it is not possible to provide a specific answer. |
| Case 2 | **Instruction**: | Considering a sensor network application, what are the necessary ESP32 module technical specifications, antenna design, and ESP-NOW protocol features, including pairing and encryption? Limit to 7 items. |
| | **Filter Response**: | No, because the question is asking for a list of specifications and features **that are too broad and would require more than 7 items** to be fully addressed. Additionally, the question combines different aspects such as technical specifications, antenna design, and protocol features, which are complex and cannot be effectively condensed into a limited list without additional context or prioritization. |

Table 6: The cases that have been filtered after the two-stage filtering, we list the original queries and why they are filtered. The filter response is generated by GPT-4 Turbo. The reasons that the query/instruction cannot be answered are highlighted.

| Model | Average | MMLU (10 shot) | ARC (25 shot) | HellaSwag (10 shot) | TruthfulQA (0 shot) | WinoGrande (5 shot) | GSM8k (5 shot) |
|---|---|---|---|---|---|---|---|
| ShareGPT | 59.68 | 59.43 | 57.34 | 81.66 | 52.30 | 71.98 | 35.41 |
| Conifer | 59.60 | 58.88 | 58.28 | 81.80 | 49.31 | 71.82 | 37.53 |

Table 7: Results on the Open LLM leaderboard.

### B.3 Results on the Open LLM Leaderboard

Several traditional studies on instruction tuning (Longpre et al., 2023; Sanh et al., 2021; Lou et al., 2024) that aggregate data from a wide array of NLP tasks to train multi-task models, mainly evaluate models on downstream NLP benchmarks. There has been a notable discrepancy shown between these NLP task performances and human preferences (Gao, 2023; Zheng et al., 2023; Dubois et al., 2024). Among them, Open LLM leaderboard is a widely used evaluation benchmark. It is important to recognize that leaderboard evaluates abilities that are less correlated with instruction-following ability of chat models (Tunstall et al., 2023b), or human preferences (Dubois et al., 2024). Although Open LLM leaderboard evaluates capabilities that are outside the primary focus of this paper, we also following previous work (Tunstall et al., 2023b), report the results to validate whether fine-tuning has introduced regressions on the model's reasoning and truthfulness capabilities.

The Open LLM Leaderboard consists of six tasks, including MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022), WinoGrande(ai2, 2019) and GSM8k (Cobbe et al., 2021). As is shown in Table 7, compared to Mistral-7B-ShareGPT baseline (ShareGPT for brevity), incorporating Conifer dataset has shown almost identical average score on the leaderboard. This outcome confirms that the inclusion of the Conifer dataset does not compromise on the model's performance on these benchmarks.

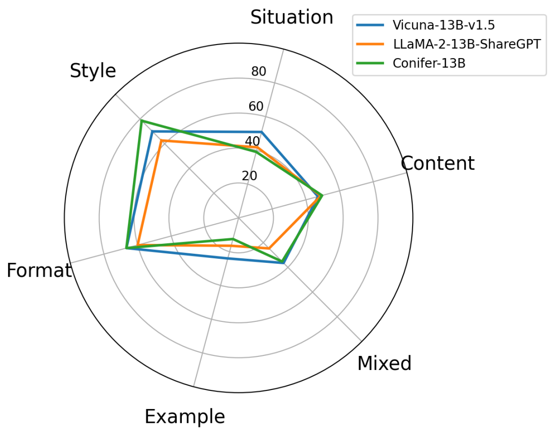
### B.4 Full Results on the IFEval and FollowBench benchmarks

We report IFEval results under the loose prompt setting and FollowBench results under the Hard Satisfaction Rate (HSR) metric in Table 2. The other metrics, including IFEval under strict prompt,

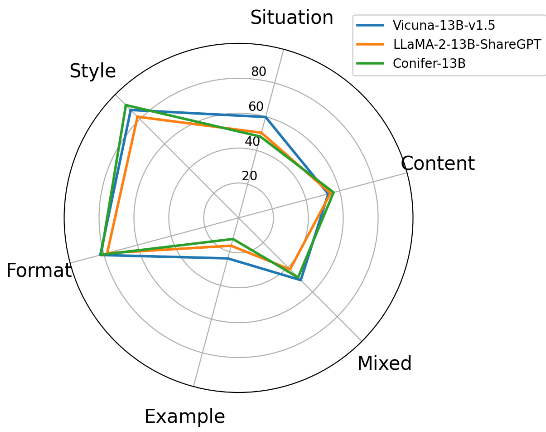
| Model | Base Model | IFEval | | | | FollowBench (SSR metric) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | strict prompt | loose prompt | strict instruction | loose instruction | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Avg |
| GPT-4† | - | 76.9 | 79.3 | 83.6 | 85.4 | 84.7 | 77.6 | 76.2 | 77.9 | 73.3 | 77.9 |
| GPT-3.5 Turbo† | - | - | - | - | - | 80.3 | 71.2 | 74.2 | 69.6 | 67.1 | 72.5 |
| Qwen-72B-Chat† | Qwen | - | 50.8 | - | - | 73.8 | 67.5 | 63.2 | 57.6 | 56.0 | 63.6 |
| LLaMA-2-70B-Chat† | LLaMA-2 | - | - | - | - | 59.9 | 57.3 | 55.7 | 53.3 | 53.2 | 55.9 |
| Vicuna-13B-v1.5 | LLaMA-2 | 43.1 | 46.6 | 53.6 | 58.0 | 69.1 | 64.5 | 58.6 | 52.6 | 52.7 | 59.5 |
| LLaMA-2-13B-ShareGPT | LLaMA-2 | 42.9 | 47.1 | 53.7 | 57.1 | 59.2 | 55.4 | 57.0 | 48.1 | 49.6 | 53.9 |
| Conifer-13B | LLaMA-2 | 42.9 | 47.5 | 53.0 | 57.4 | 60.5 | 58.3 | 58.2 | 53.9 | 51.1 | 56.4 |
| Deita-7B-v1.0-SFT | Mistral | 42.0 | 45.1 | 54.3 | 57.3 | 55.8 | 58.5 | 54.0 | 47.6 | 51.4 | 53.5 |
| Zephyr-7B-beta | Mistral | 32.0 | 44.9 | 46.8 | 58.0 | 57.6 | 55.9 | 54.2 | 54.3 | 48.5 | 54.1 |
| Deita-7B-v1.0 | Mistral | 44.6 | 51.9 | 56.6 | 63.7 | 55.8 | 54.8 | 54.7 | 52.4 | 53.7 | 54.3 |
| Mistral-7B-Muffin | Mistral | 32.9 | 34.0 | 44.0 | 45.4 | 40.1 | 39.7 | 37.9 | 35.7 | 36.7 | 38.0 |
| Mistral-7B-Evol-Instruct | Mistral | 41.4 | 44.0 | 51.3 | 54.4 | 53.2 | 57.0 | 54.7 | 51.2 | 47.4 | 52.7 |
| Mistral-7B-ShareGPT | Mistral | 37.5 | 43.4 | 49.3 | 54.9 | 55.7 | 56.6 | 53.6 | 53.4 | 49.7 | 53.8 |
| Conifer-7B | Mistral | 45.8 | 50.8 | 57.1 | 62.0 | 53.9 | 57.6 | 53.7 | 54.5 | 49.7 | 53.9 |
| Mistral-7B-ShareGPT-DPO | Mistral | 43.8 | 48.2 | 55.8 | 59.7 | 58.4 | 58.1 | 55.8 | 54.9 | 52.3 | 55.9 |
| Conifer-7B-DPO | Mistral | 48.1 | 52.3 | 59.1 | 63.3 | 60.3 | 55.7 | 55.7 | 55.9 | 53.3 | 56.2 |

Table 8: Additional results on two instruction following benchmarks: IFEval and FollowBench. † indicates that the results are directly sourced from the original benchmarks.

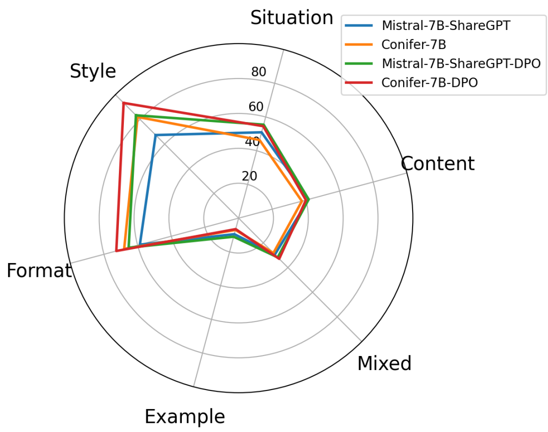
loose instruction and strict instruction settings, and FollowBench under Soft Satisfaction Rate (SSR) metric are detailed in Table 8. We also illustrate the FollowBench scores on the contained six categories as shown in Figure 6. These metric results are in agreement with the results shown in Table 2.



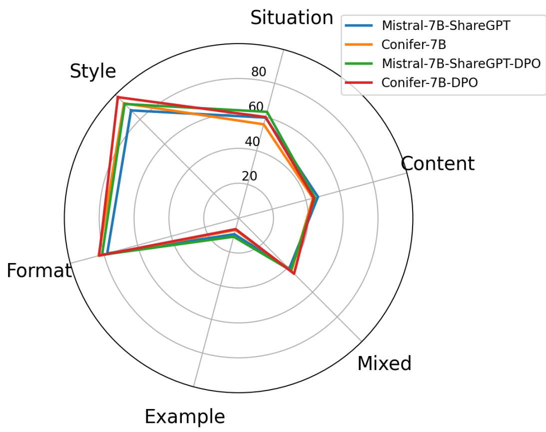

(a) LLaMA HSR




(b) LLaMA SSR




(c) Mistral HSR




(d) Mistral SSR

Figure 6: Average HSR (%) and SSR (%) results across different levels in diverse constraint categories of our models and baselines.