EIFBENCH: Extremely Complex Instruction Following Benchmark for Large Language Models

Anonymous ACL submission

Abstract

In the advancement of large language models (LLMs), while there have been notable improvements in their ability to generalize across various natural language processing tasks, existing datasets often lack the complexity required to fully reflect real-world scenarios. These datasets predominantly focus on single-task environments with limited constraints, thereby failing to capture the multifaceted and constraint-rich requirements inherent in practical applications. To bridge this gap, we present the extremely complex instruction following benchmark (EIFBENCH), meticulously crafted to facilitate a more realistic and robust evaluation of LLMs. EIFBENCH offers several distinctive advantages: Firstly, it includes multi-task scenarios that enable comprehensive assessment across diverse task types concurrently. Secondly, it is sourced from a wide array of diverse origins to ensure both the diversity and representativeness of its data. Lastly, it integrates a variety of constraints, replicating complex operational environments and providing critical insights into the models' capabilities under resource, time, and environmental limitations. Evaluations on EIFBENCH have unveiled considerable performance discrepancies in existing LLMs when challenged with these extremely complex instructions. This finding underscores the necessity for ongoing optimization and the development of more versatile and deeply understanding models, equipped to navigate the intricate challenges posed by real-world applications.

1 Introduction

The advent of large-scale language models has transformed real-world applications by enhancing machines' ability to comprehend a diverse range of human instructions, from simple conversations to complex problem-solving (Sanh et al., 2022; Dubois et al., 2023). Thus, instructions have become central to effective human-machine interaction in this new landscape (Zhong et al., 2021; Mishra et al., 2022; Gao et al., 2024). However, as user demands grow more sophisticated, traditional benchmarks (Zhong et al., 2024; Chia et al., 2023), which focus on specific tasks, are insufficient to evaluate models' comprehensive ability to handle multifaceted instructions. This shortfall underscores the need for innovative evaluation frameworks capable of accurately assessing how models understand and execute complex instructions (Zhou et al., 2023; Wang et al., 2023; Xu et al., 2024).

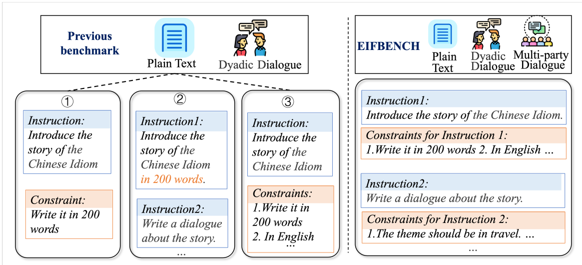

Figure 1: Previous benchmarks vs. EIFBENCH

To evaluate the instruction following abilities of LLMs, several benchmarks (Zhou et al., 2023; Qin et al., 2024; Li et al., 2024) have been proposed, which can be categorized into three main types. ***Single-instruction single-constraint*** benchmarks, such as IFEval (Zhou et al., 2023) and INFOBENCH (Qin et al., 2024), focus on tasks governed by a single constraint, providing insights into basic instruction following abilities. In contrast, ***single-instruction multi-constraint*** benchmarks, like CFBench (He et al., 2024b), evaluate how models handle a single instruction with multiple constraints across content, numerical, and other dimensions simultaneously. Additionally, ***multi-instruction single-constraint*** scenarios, such as those explored by SIFo (Chen et al., 2024), test models' adherence to sequences of instructions, assessing their adaptability and versatility while maintaining focus on a single constraint. Nonetheless, there remains a gap in research addressing ***multi-instruction multi-constraint*** scenarios, which more accurately reflect real-world complexities.

Nevertheless, multi-instruction multi-constraint (MIMC) scenarios are ubiquitous in real-world applications, such as workflow automation (Zhang et al., 2022; Taylor et al., 2023) and healthcare scheduling (Bakhshandeh and Al-e-hashem, 2024; Li et al., 2021). For example, in cloud-based workflow automation, orchestrating computational tasks such as data preprocessing, model inference, and report generation requires balancing resource allocation, execution time, and task dependencies (Xiong et al., 2016). However, existing LLMs struggle with such complexity, with performance dropping by over 30% with over 5 constraints (He et al., 2024b). Bridging this gap necessitates benchmarks that mirror real-world MIMC dynamics, integrating both task interdependence and constraint scalability to foster robust and adaptable LLMs.

In response to these challenges, we introduce the extremely complex instruction following benchmark (**EIFBENCH**), specifically designed to address the shortcomings of current evaluation datasets by providing a comprehensive framework that mirrors the complexities of real-world task environments. EIFBENCH is unique in its inclusion of multi-task scenarios, drawn from diverse sources and integrated with multifaceted constraints, as shown in Fig. 1[1]. This design allows for an in-depth assessment of a model's ability to manage complex demands and adapt dynamically to various operational parameters. The main contributions of this paper are summarized as follows:

- We first present the extremely complex instruction following benchmark (EIFBENCH), which addresses the current gap in NLP research for evaluating model generalization in complex multi-task environments.
- EIFBENCH comprises 1,000 crafted data samples across three scenarios, simulating real-world applications with multiple instructions and constraints.
- We perform a categorized analysis of 13 LLMs, including open-source, closed-source, and reasoning models (e.g., DeepSeek-R1), revealing their limitations in handling complex instructions and identifying directions for improvement in adapting to real-world complex contexts.

[1] In this work, plain text datasets refer to non-conversational plain text datasets.

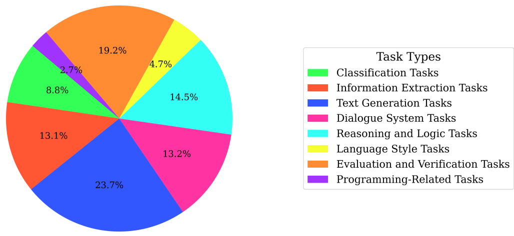

Figure 2: Task type distribution in EIFBENCH.

2 EIFBENCH Framework

2.1 Overview

To thoroughly assess the capability of large language models (LLMs) in adhering to complex instructions, we introduce an exceptionally challenging instruction following benchmark. Specifically, we categorize both tasks and constraints to structure the evaluation. For tasks, we identify and compile 8 types of tasks based on traditional NLP tasks. Regarding constraints, we establish a two-level hierarchical taxonomy for the organization.

2.2 Task Categories

In line with existing works instruction following (Zhang et al., 2024a; Li et al., 2024), we categorize the tasks in EIFBENCH into eight primary types. These categories provide a comprehensive framework for systematically evaluating model performance across diverse task settings. The distribution of these task categories within EIFBENCH allows for a thorough analysis, as shown in Fig. 2.

Classification Tasks encompass a variety of classification needs. Basic tasks include sentiment analysis, text classification based on themes or types, and toxic content detection. Advanced classification tasks extend to empathy detection, argument mining, gender/personality trait classification, stereotype detection, and social norm judgment.

Information Extraction Tasks involve extracting and organizing key information from text. Representative tasks include named entity recognition (NER), keyword annotation, coreference resolution, and entity relationship classification.

Text Generation Tasks cover both creative and practical applications, which include story and poetry generation, recipe creation, text expansion/compression, headline generation, data description generation and so on.

Dialogue System Tasks are designed for developing interactive dialogue agents like dialogue generation, intent recognition, question generation/rewriting, dialogue state tracking, and role-playing dialogues.

Reasoning and Logic Tasks require models to demonstrate logical inference and critical thinking. Tasks include commonsense question answering, multi-hop question answering, critical thinking assessment, mathematical reasoning, and theory of mind reasoning.

Language Style Tasks focus on the manipulation and analysis of language styles. Tasks in this category include style transfer, language feature analysis, sarcasm detection, and the identification of spelling and punctuation errors.

Evaluation and Verification Tasks involve the verification of information and assessment of text quality such as text quality assessment, fact verification and answer validation.

Programming-Related Tasks are designed to evaluate the model's understanding and synthesis of programming languages. Tasks include code generation, debugging, and code explanation.

Furthermore, tasks are organized into distinct structural modes: parallel task mode for simultaneous consideration of multiple dimensions, serial task mode for chain-dependent tasks, conditional selection mode which adapts based on varying conditions, and nested task mode for tasks embedded within hierarchical structures. These categorizations enable a systematic and comprehensive evaluation of model capabilities within the benchmark.

2.3 Constraint Categories

Following established research on instruction following (Zhang et al., 2024b), we have developed a comprehensive constraint system for EIFBENCH. This system categorizes constraints into four primary types: Content Constraints, Situation Constraints, Style Constraints, and Format Constraints. These categories provide a structured framework to systematically evaluate the capabilities of language models across a wide range of instructional scenarios. The distribution is shown in Fig. 3. Detailed descriptions of the specific constraint dimensions within each category are provided in Appendix A.

1. **Content Constraints**: These constraints focus on the thematic and informative content requirements, ensuring that the generated text adheres to specific topics, inclusion/exclusion criteria, values, privacy considerations, and numerical precision.

2. **Situation Constraints**: These constraints emphasize the context and role-playing aspects of content generation, targeting audience specifications, preconditions, and the integration of various background information formats.

3. **Style Constraints**: These constraints govern the stylistic and emotional aspects of the generated text, including tone, emotion, linguistic characteristics, and multilingual capabilities.

4. **Format Constraints**: These constraints ensure that the output adheres to specific structural and formatting requirements, including output formats, text patterns, grammatical structures, citations, numbering, hierarchical organization, and template adherence.

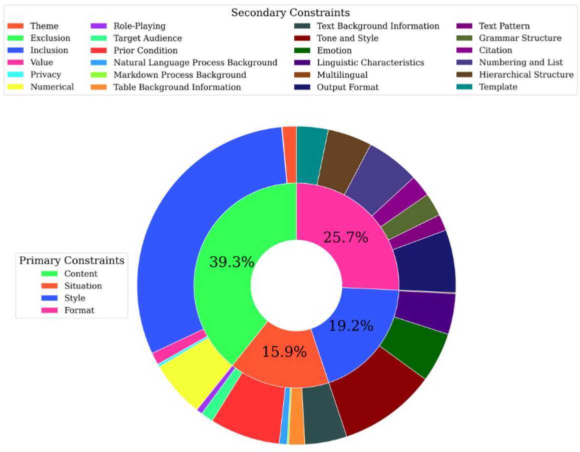

Figure 3: Constraint type distribution in EIFBENCH.

3 EIFBENCH Construction

This section describes the construction pipeline of EIFBENCH and the evaluation protocol.

3.1 Data Collection

The overall construction process includes several key stages: 1) Taxonomy of Constraints and Tasks, 2) Multi-scenario Data Collection, 3) Task Expansion, 4) Constraint Expansion, 5) Quality Control, and 6) Response Generation & Evaluation.

Taxonomy of Constraints and Tasks. We establish two taxonomies for constraints and tasks, as presented in Section 2.

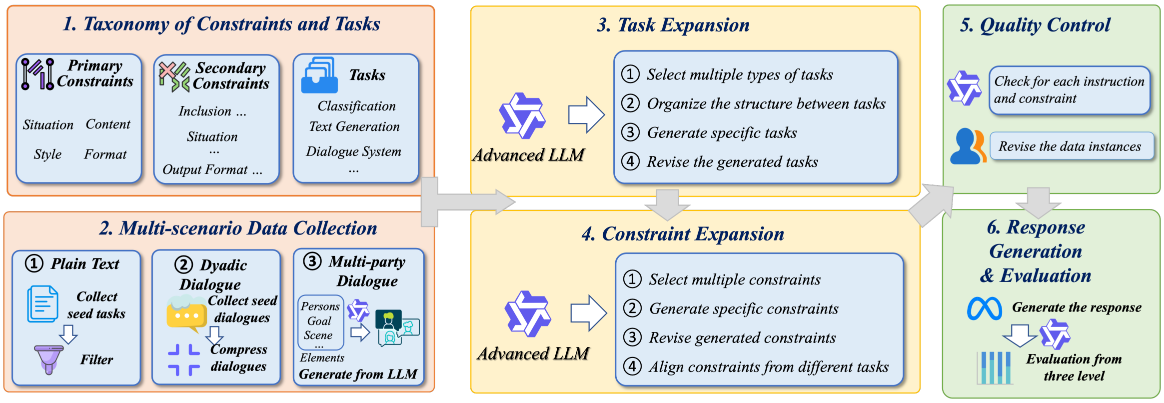

Figure 4: Pipeline of constructing the benchmark.

Multi-scenario Data Collection. Our data collection process encompasses three distinct types of seed instruction datasets: plain text, dyadic dialogue, and multi-party dialogue. For the plain text dataset, we directly source examples from existing literature (Wen et al., 2024; Li et al., 2024). The dyadic dialogue dataset is compiled from real-life interactions, followed by data cleaning and noise reduction processes. Given that dialogues are often lengthy, we utilize large language models (LLMs) to condense these conversations while preserving core information. For the multi-party dialogue dataset, we generate data using LLMs by setting diverse scene scenarios and varying the number of participants. We craft specific prompts to have the LLM produce diverse and representative multi-party dialogue data.

Task Expansion. In the plain text scenario, individual tasks are expanded into a series of tasks (detailed in Section 2.2). During task generation, we leverage LLMs to create task sets with complex structures, such as dependent relationships and parallelism among tasks. Concurrently, we conduct rigorous task quality assessments, removing redundant tasks, those beyond model capabilities, and contradictory tasks, thus ensuring the generated data's quality and consistency. In dyadic dialogue and multi-party dialogue scenarios, we directly generate multiple new tasks, ensuring each task reflects the complexity of real-world interactions.

Constraint Expansion. In the constraint expansion process, we refine and complexify simple instructions based on a predefined constraint taxonomy (refer to Section 2.3). By utilizing LLMs, we incrementally add complexity to the instructions, ensuring that each task spans a broad spectrum of operational requirements and constraints. Throughout this process, we iteratively review and revise the constraints, specifically targeting and clarifying those with ambiguous semantics, to ensure that all constraints could be objectively evaluated and quantified. This approach not only increases the tasks' complexity and challenge but also enhances the realism and comprehensiveness of the generated data.

Quality Assessment. Our quality assessment involves instruction-level validation and constraint-level validation. For instruction-level validation, we analyze relationships between instructions, ensuring logical consistency and feasibility for LLM models, removing contradictory, redundant, or infeasible tasks while maintaining a diverse and moderate difficulty task set of 6 to 12 instructions. For constraint-level validation, we iteratively refine constraints based on predefined taxonomies, ensuring they can be objectively quantified and are within model capabilities, modifying any constraints that are ambiguous or beyond the model's capability to complete.

Response Generation & Evaluation. First, using the instruction data, we employ various language models to generate the corresponding outputs. To verify their compliance, we then prompt large language models to assess each constraint satisfaction for the outputs, generating a binary outcome (0/1) that indicates whether the generated output satisfies the respective constraints.

3.2 Dataset Statistics

As shown in Table 1, EIFBENCH consists of 1,000 instances for evaluation. Across the three datasets, the minimum and maximum numbers of constraints per instance are provided, with average numbers outlined as well. Fig. 5 and Table 2 illustrate the distribution of constraint numbers and the distri-

Category	#N	Min.	Max.	Avg.
Plain Text	450	41	107	73.27
Dyadic Dialogue	450	47	107	73.38
Multi-party Dialogue	100	63	116	80.26

Table 1: Statistics for Plain Text, Dyadic Dialogue, and Multi-party Dialogue. #N is the number of data instances; Min., Max., and Avg. mean the minimum, maximum, and average number of constraints per instance.

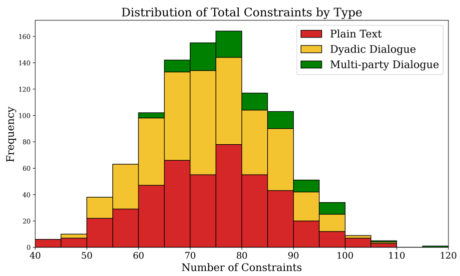

Figure 5: Distributions of total constraints for different text categories.

Scenario	6	7	8	9	10	11	12
Plain Text	15	76	136	139	76	7	1
Dyadic Dialogue	42	113	152	108	33	2	0
Multi-party Dialogue	0	11	47	27	13	1	1

Table 2: Distributions of instructions with different number of constraints.

bution of instruction numbers within EIFBENCH, respectively.

3.3 Evaluation Protocol

We employ `Qwen2.5-72B-Instruct` as the evaluation model to assess constraint adherence in generated responses. Following established practices (Wen et al., 2024), each constraint is assigned a binary compliance score $S_{i,j,k} \in \{0, 1\}$, where 1 indicates full adherence.

Global Accuracy (GAcc) evaluates strict end-to-end task success, requiring *all* constraints across *all* instructions to be satisfied simultaneously. This metric reflects real-world scenarios where partial compliance is insufficient (e.g., legal document generation requiring 100% constraint adherence):

$$\text{GAcc} = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{m_i}\prod_{k=1}^{c_{i,j}} S_{i,j,k} \quad (1)$$

where n denotes the number of total instances, m_i is the number of instructions in the instance i, and $c_{i,j}$ is the number of constraints in the instruction j of instance i.

Instruction-Level Accuracy (ILAcc) quantifies the per-instruction success rate by averaging compliance across instructions for a single instance. It identifies fragile components in multi-step workflows (e.g., failed data parsing steps in analytics pipelines):

$$\text{ILAcc} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\left(\frac{1}{c_{i,j}}\prod_{k=1}^{c_{i,j}} S_{i,j,k}\right) \quad (2)$$

Constraint-Level Accuracy (CLAcc) assesses atomic constraint fulfillment, crucial for debugging specific requirement violations (e.g., detecting which safety constraints fail in robot control commands):

$$\text{SCLAcc} = \frac{1}{\sum_{j=1}^{m_i} c_{i,j}}\sum_{j=1}^{m_i}\sum_{k=1}^{c_{i,j}} S_{i,j,k} \quad (3)$$

$$\text{CLAcc} = \frac{1}{n}\sum_{i=1}^{n}\text{SCLAcc} \quad (4)$$

These metrics progressively assess compliance at different granularities: from strict task-level requirements (GAcc) through intermediate instruction-level compliance (ILAcc) to fine-grained constraint-level analysis (CLAcc).

3.4 Evaluation Set Quality

To generate high-quality evaluation data, we implement a post-inspection protocol following the initial generation. First, we employ `Qwen2.5-72B-Instruct` to systematically validate instruction-clarity alignment, constraint logical consistency, and task feasibility, while automatically identifying and rectifying detectable errors through iterative self-correction. Subsequently, three certified labeling specialists conduct manual inspection to eliminate redundant constraints and instructions, revise infeasible tasks, and resolve ambiguous formulations, ensuring both technical rigor and practical executability.

4 Experiments

4.1 Baselines

We compare the performance of both proprietary and open-source LLMs trained on diverse corpora. In the proprietary category, we evaluate models such as GPT-4o (OpenAI, 2023), GPT-4o-mini (OpenAI, 2023), Claude3.5-Sonnect (Anthropic,

Model	GAcc	ILAcc	CLAcc
Closed-Source Models			
GPT-4o	0.0156	0.3366	0.7501
Claude-3.5-Sonnect	0.0022	0.2542	0.7004
GPT-4o-mini	0.0067	0.3244	0.7573
Claude-3.5-Haiku	0.0000	0.1125	0.4560
Open-Source Models			
LlaMA3.1-8B-Instruct	0.0000	0.0994	0.5522
LlaMA3.1-70B-Instruct	0.0000	0.2125	0.7377
Qwen2-7B-Instruct	0.0000	0.2179	0.6333
Qwen2-72B-Instruct	0.0044	0.4071	0.8443
gemini-1.5-Pro	0.0133	0.4400	0.8083
Qwen2.5-7B-Instruct	0.0044	0.3574	0.8440
Qwen2.5-72B-Instruct	0.0200	0.5514	0.8996
Reasoning Models			
DeepSeek-R1	0.0444	0.5725	0.8989
QwQ-32B-Preview	0.0111	0.3102	0.6102

Table 3: Performance metrics for plain text tasks.

Model	GAcc	ILAcc	CLAcc
Closed-Source Models			
GPT-4o	0.0178	0.3809	0.6904
Claude-3.5-Sonnect	0.0022	0.2511	0.5896
GPT-4o-mini	0.0022	0.2767	0.7266
Claude-3.5-Haiku	0.0000	0.1203	0.3849
Open-Source Models			
LlaMA3.1-8B-Instruct	0.0000	0.1008	0.5618
LlaMA3.1-70B-Instruct	0.0000	0.1747	0.6941
Qwen2-7B-Instruct	0.0000	0.2020	0.6222
Qwen2-72B-Instruct	0.0022	0.3908	0.8296
gemini-1.5-Pro	0.0178	0.4334	0.8424
Qwen2.5-7B-Instruct	0.0022	0.2520	0.7808
Qwen2.5-72B-Instruct	0.0180	0.5189	0.9014
Reasoning Models			
DeepSeek-R1	0.0622	0.5537	0.8411
QwQ-32B-Preview	0.0133	0.3489	0.6475

Table 4: Performance metrics for dyadic dialogue tasks.

2024b), and Claude3.5-Haiku (Anthropic, 2024a). These models are designed to demonstrate advanced language processing capabilities. Among open-source models, we assess LlaMA3.1 (Dubey et al., 2024), Qwen2 (Yang et al., 2024a), gemini-1.5-Pro (Reid et al., 2024), and Qwen2.5 (Yang et al., 2024b), which have been trained extensively on multilingual data. Additionally, we include reasoning models like DeepSeek-R1 (Reid et al., 2024) and QwQ-32B-Preview[2] to explore their efficiency.

4.2 Settings

For inference, we handle proprietary models by accessing their APIs, ensuring efficient processing. For open-source models, we utilize a setup of four Nvidia A100 GPUs, each with 80GB of VRAM, leveraging the vLLM framework on EIFBENCH where applicable. This configuration allows the completion of all tasks in approximately 30 minutes. During the evaluation phase, which spans 4 to 18 hours depending on task complexity, we deploy the same four Nvidia A100 GPUs setup across all models. The `Qwen2.5-72B-Instruct` model serves as the evaluator, providing comprehensive assessment capabilities for model performance.

4.3 Results Analysis

4.3.1 Is EIFBENCH Challenging?

The EIFBENCH evaluation, as reflected in Tables 3, 4, and 5, presents multifaceted challenges to language models by mirroring real-life scenarios through three distinct datasets: plain text tasks, dialogue tasks, and multi-party dialogue tasks. These datasets each embody unique aspects of real-world applications. These datasets capture distinct aspects of practical applications, with plain text tasks focusing on straightforward information processing, dyadic dialogue tasks reflecting the dynamics of conversational interactions, and multi-party dialogues simulating collaborative discussions.

Our evaluation uses three key metrics: Global Accuracy (GAcc), Instruction-Level Accuracy (ILAcc), and Constraint-Level Accuracy (CLAcc). Constraint-level evaluation, widely emphasized in recent studies (Zhang et al., 2024a,b; Li et al., 2024), focuses on individual constraint execution. This metric typically yields the highest accuracy, demonstrating models' capability to handle isolated constraints effectively. However, instruction-level performance shows a significant decline. ILAcc evaluates the accuracy of completing individual *instructions* and reveals models' difficulty in fulfilling *all* constraints per instruction. Despite high CLAcc scores, models struggle to satisfy all constraints required for *an instruction*. Consequently, the probability of successfully executing *all instructions* within an instance remains low, highlighting the need for improved model capabilities in complex multi-task scenarios. In real-world contexts, LLMs need to enhance their ability to fully adhere to all instructions in comprehensive task execution, highlighting the challenging nature of the

[2]https://modelscope.cn/models/Qwen/QwQ-32B-Preview

Model	GAcc	ILAcc	CLAcc
Closed-Source Models			
GPT-4o	0.0000	0.3021	0.6569
Claude-Sonnect	0.0000	0.2236	0.5728
GPT-4o-mini	0.0000	0.3187	0.7667
Claude-Haiku	0.0000	0.0728	0.2589
Open-Source Models			
LlaMA3.1-8B-Instruct	0.0000	0.0856	0.5527
LlaMA3.1-70B-Instruct	0.0000	0.1360	0.7139
Qwen2-7B-Instruct	0.0000	0.1326	0.6288
Qwen2-72B-Instruct	0.0000	0.3266	0.8234
gemini-1.5-Pro	0.0100	0.3959	0.8538
Qwen2.5-7B-Instruct	0.0000	0.2596	0.8315
Qwen2.5-72B-Instruct	0.0000	0.5546	0.9270
Reasoning Models			
DeepSeek-R1	0.0100	0.4883	0.8712
QwQ-32B-Preview	0.0200	0.3075	0.6892

Table 5: Performance metrics for multi-party dialogues.

Model	Step	GAcc	ILAcc	CLAcc
LlaMA3.1-8B	1	0.0000	0.0856	0.5527
	2	0.0000	0.0634	**0.5593** ↑
Qwen2-7B	1	0.0000	0.1326	0.6288
	2	0.0000	**0.1484** ↑	**0.6611** ↑
Qwen2.5-7B	1	0.0000	0.2596	0.8315
	2	0.0000	**0.2748** ↑	0.8087

Table 6: Performance of different generation times on multi-party dialogues. Model is its *Instruction* version.

EIFBENCH dataset.

Model performance varies significantly across categories, with task-type dependency observed. Among closed-source models, GPT-4o shows situational superiority: it leads in dyadic dialogues but fails completely in multi-party scenarios. For open-source models, gemini-1.5-Pro demonstrates competitive constraint-level accuracy, outperforming most open-source models except Qwen2.5-72B-Instruct. Qwen2.5-72B-Instruct dominates in three scenarios since we generate data with it. The poor performance of LlaMA3.1 models may be attributed to architectural incompatibility with constraint-chained scenarios. Contrary to the reasoning models, DeepSeek-R1 outperforms others in plain text tasks and dyadic dialogue tasks on ILAcc. These patterns emphasize that model effectiveness depends on both base capability and structural alignment with task hierarchies.

4.3.2 Factors Influencing Instruction Following

Our investigation identifies two critical dimensions influencing instruction adherence in language models: (1) the number of instructions per instance and (2) the number of constraints per instruction. As illustrated in Fig. 6, performance degrades progressively as these variables increase, though with minor patterns. This decline is particularly pronounced with an increase in constraints, likely because each additional constraint raises the complexity of completing a task, making it more challenging for the model to meet all requirements. Conversely, the interdependence between instructions is generally low, meaning that an increase in the number of instructions does not lead to as steep a performance decline. This is primarily because the difficulty lies in managing multiple tasks simultaneously, rather than the instructions themselves being interrelated. In some instances, especially where there are larger numbers of instructions and constraints, performance may inexplicably improve. This can be attributed to the smaller sample sizes in these scenarios, leading to greater variability in performance outcomes. Overall, this analysis underscores the intricacies of maintaining consistent instruction adherence across diverse scenarios.

4.3.3 Analysis of Further Thinking

The results presented in Table 6 demonstrate the impact of a two-step generation approach on the performance of smaller open-source models in multi-party dialogue tasks. By generating an initial output and then refining it through further reflection, these models show consistent improvements across most scenarios. These findings suggest that iterative reasoning allows smaller models to identify and correct errors, enhancing their problem-solving capabilities without requiring larger architectures. It underscores the value of fostering deeper reasoning strategies in smaller models, enabling them to achieve higher accuracy in complex tasks.

5 Related work

5.1 Instruction Following

Recent advancements in the fine-tuning of large language models (LLMs) have demonstrated the significant impact of annotated instructional data on enhancing models' ability to understand and execute a wide array of language instructions (Weller et al., 2020; Ye and Ren, 2021; Mishra et al., 2022). Building on this, incorporating more detailed and

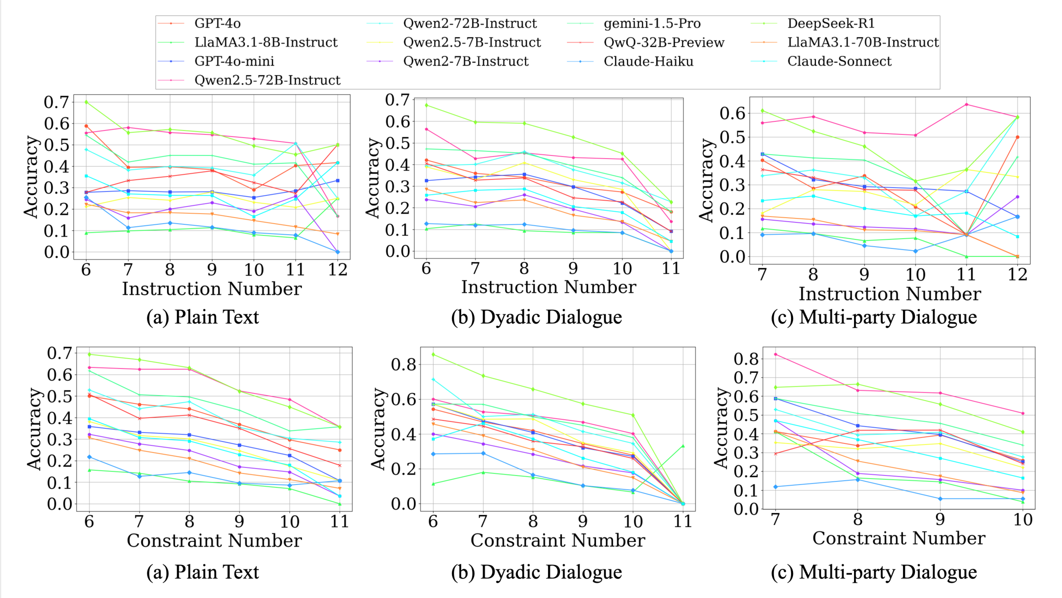

Figure 6: Performance on different number of instructions and constraints.

sophisticated instructions has been shown to further improve model capabilities (Lou et al., 2023). For example, the study by (Xu et al., 2024) introduces a method where complex instructions are incrementally generated from seed instructions using LLMs, resulting in fine-tuning that allows LLaMA to achieve performance exceeding 90% of ChatGPT's capacity across 17 out of 29 skills. Additionally, the research community is increasingly focused on constrained instructions (Sun et al., 2024; Dong et al., 2024; He et al., 2024a), a subset of complex instructions, which involves enhancing instructional complexity by increasing the number of constraints, thereby improving the models' ability to handle intricate tasks.

5.2 Evaluation of Instruction Following

Instruction following is a pivotal aspect influencing the effectiveness of large language models (LLMs) (Liu et al., 2023). Early work focused on evaluating compliance with simple human directives, typically featuring single constraints, such as semantic (Zheng et al., 2023; Liu et al., 2024) or formatting (Xia et al., 2024; Tang et al., 2024) requirements. As LLMs are increasingly applied in complex real-world contexts, there is a growing need to assess their ability to handle intricate instructions (Qin et al., 2024; Jiang et al., 2024). For instance, (Sun et al., 2024) introduced the Conifer dataset along with a progressive learning framework to bolster LLMs' abilities to process multi-level instructions featuring complex constraints. Meanwhile, (Qin et al., 2024) proposed a method for breaking down a singular instruction into multiple constraints. (He et al., 2024b) curated constraints derived from real-world contexts to construct an advanced benchmark, which utilizes comprehensive task descriptions and inputs to evaluate LLMs. Furthermore, (Wen et al., 2024) developed an innovative benchmark by integrating and enhancing data from existing datasets, emphasizing combinations of diverse constraint types. Nonetheless, we contend that existing datasets suffer from a limited number of constraints and primarily focus on single-instruction scenarios, whereas real-world applications often involve multi-instruction, multi-constraint instructions where the number of constraints greatly exceeds those found in current datasets.

6 Conclusion

In conclusion, the Extremely Complex Instruction Following Benchmark (EIFBENCH) addresses the limitations of existing datasets by introducing multi-task scenarios, diverse data sources, and complex constraints, enabling a more realistic evaluation of large language models (LLMs). Evaluations on EIFBENCH reveal significant performance gaps in current LLMs, underscoring the need for further optimization and the development of models capable of handling real-world complexities. This benchmark sets a new standard for future research, driving the creation of more robust and adaptable systems for practical applications.

7 Limitations

While EIFBENCH provides a robust evaluation framework for plain text, dyadic dialogue, and multi-party tasks, it has two limitations that could be addressed in future work. First, the inter-task relationships could be further enhanced to reflect more complex, real-world dependencies, such as multi-step reasoning or conditional task execution. Second, the dataset currently focuses primarily on Chinese instructions, which limits its applicability to multilingual scenarios. Expanding to include more languages would improve its global relevance and enable evaluation of LLMs' cross-lingual capabilities. Addressing these limitations would make EIFBENCH even more comprehensive and aligned with practical applications.

References

AI Anthropic. 2024a. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

AI Anthropic. 2024b. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3(6).

Azam Bakhshandeh and Seyed Mohammad Javad Mirzapour Al-e-hashem. 2024. A multi-objective scheduling model in medical tourism centers considering multi-task staff training. *Eng. Appl. Artif. Intell.*, 131:107808.

Xinyi Chen, Baohao Liao, Jirui Qi, Panagiotis Eustratiadis, Christof Monz, Arianna Bisazza, and Maarten de Rijke. 2024. The sifo benchmark: Investigating the sequential instruction following ability of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1691–1706. Association for Computational Linguistics.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. INSTRUCTEVAL: towards holistic evaluation of instruction-tuned large language models. *CoRR*, abs/2306.04757.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Self-play with execution feedback: Improving instruction-following capabilities of large language models. *CoRR*, abs/2406.13542.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jie Gao, Simret Araya Gebreegziabher, Kenny Tsu Wei Choo, Toby Jia-Jun Li, Simon Tangi Perrault, and Thomas W. Malone. 2024. A taxonomy for human-llm interaction modes: An initial exploration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA 2024, Honolulu, HI, USA, May 11-16, 2024*, pages 24:1–24:11. ACM.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. 2024a. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 10864–10882. Association for Computational Linguistics.

Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024b. Can large language models understand real-world complex instructions? In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18188–18196. AAAI Press.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4667–4688. Association for Computational Linguistics.

Yizhi Li, Ge Zhang, Xingwei Qu, Jiali Li, Zhaoqun Li, Noah Wang, Hao Li, Ruibin Yuan, Yinghao Ma, Kai Zhang, Wangchunshu Zhou, Yiming Liang, Lei Zhang, Lei Ma, Jiajun Zhang, Zuowen Li, Wenhao Huang, Chenghua Lin, and Jie Fu. 2024. Cif-bench: A chinese instruction-following benchmark for evaluating the generalizability of large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 12431–12446. Association for Computational Linguistics.

Yong Li, Xuan-Yu Jiao, Bai-Qing Sun, Qiu-Hao Zhang, and Junyou Yang. 2021. Multi-welfare-robot cooperation framework for multi-task assignment in healthcare facilities based on multi-agent system. In *IEEE International Conference on Intelligence and Safety for Robotics, ISR 2021, Tokoname, Japan, March 4-6, 2021*, pages 413–416. IEEE.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. Alignbench: Benchmarking chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11621–11640. Association for Computational Linguistics.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *CoRR*, abs/2308.05374.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2023. A comprehensive survey on instruction following. *arXiv preprint arXiv:2303.10475*.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3470–3487. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 13025–13048. Association for Computational Linguistics.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Baohua Dong, Ran Lin, and Ruohui Huang. 2024. Conifer: Improving complex constrained instruction-following ability of large language models. *CoRR*, abs/2404.02823.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 12–34. Association for Computational Linguistics.

Connor J Taylor, Kobi C Felton, Daniel Wigh, Mohammed I Jeraal, Rachel Grainger, Gianni Chessari, Christopher N Johnson, and Alexei A Lapkin. 2023. Accelerated chemical reaction optimization using multi-task learning. *ACS Central Science*, 9(5):957–968.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1361–1375. Association for Computational Linguistics.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxing Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Benchmarking complex instruction-following with multiple constraints composition.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. FOFO: A benchmark to evaluate llms' format-following capability. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 680–699. Association for Computational Linguistics.

Fu Xiong, Cang Yeliang, Zhu Lipeng, Hu Bin, Deng Song, and Wang Dong. 2016. Deadline based scheduling for data-intensive applications in clouds. *The Journal of China Universities of Posts and Telecommunications*, 23(6):8–15.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *CoRR*, abs/2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Qinyuan Ye and Xiang Ren. 2021. Learning to generate task-specific adapters from task description. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 646–653. Association for Computational Linguistics.

Lijun Zhang, Xiao Liu, and Hui Guan. 2022. Automtl: A programming framework for automating efficient multi-task learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tao Zhang, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Tao Zhang, Fan Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, Wentao Zhang, and Zenan Zhou. 2024a. Cfbench: A comprehensive constraints-following benchmark for llms. *CoRR*, abs/2408.01122.

Xinghua Zhang, Haiyang Yu, Cheng Fu, Fei Huang, and Yongbin Li. 2024b. IOPO: empowering llms with complex instruction following via input-output preference optimization. *CoRR*, abs/2411.06208.

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 *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana,* *Dominican Republic, 16-20 November, 2021*, pages 2856–2878. Association for Computational Linguistics.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 2299–2314. Association for Computational Linguistics.

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

A Taxonomy of Constraint

Constraint Type	**Constraint Dimension**
Content Constraint	Theme Constraint
	Exclusion Constraint
	Inclusion Constraint
	Value Constraint
	Privacy Constraint
	Numerical Constraint
Situation Constraint	Role-Playing Constraint
	Target Audience Constraint
	Prior Condition Constraint
	Natural Language Process Background Information Constraint
	Markdown Process Background Information Constraint
	Table Background Information Constraint
	Text Background Information Constraint
Style Constraint	Tone and Style Constraint
	Emotion Constraint
	Linguistic Characteristics Constraint
	Multilingual Constraint
Format Constraint	Output Format Constraint
	Text Pattern Constraint
	Grammar Structure Constraint
	Citation Constraint
	Numbering and List Constraint
	Hierarchical Structure Constraint
	Template Constraint

Table 7: Constraints and Their Dimensions