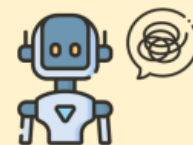




AgentIF



AGENTIF: Benchmarking Instruction Following of Large Language Models in Agentic Scenarios

Yunjia Qi*, Hao Peng*, Xiaozhi Wang, Youfeng Liu, Bin Xu, Lei Hou, Juanzi Li
Tsinghua University, Zhipu Ai



清华大学
Tsinghua University



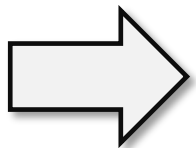
Code



Paper

Motivation

- Large Language Models (LLMs) have demonstrated advanced capabilities in real-world agentic applications.
- Agentic scenarios often involve lengthy instructions with complex constraints, such as extended system prompts and detailed tool specifications.
- Following these instructions is essential for task success and demonstrates a core capability of LLMs.
- However, whether LLMs can effectively follow instructions in real-world agentic scenarios remains underexplored.



**AGENTIF: Benchmarking Instruction Following of
Large Language Models in Agentic Scenarios**

An Example of AGENTIF

You are a top-tier code expert.

Your objective: use specified predefined functions to write structured, high-quality Python code for <task>. [C1]



Task Description

1. Task Structure

- <task>: The user-provided task requiring code. When implementing, consider preceding code and execution history. [C2]
- If <task> has multiple objectives, complete only the first. [C3]
- Output comments and code within <code></code> tags. Call the given predefined function at most once. [C5]
- If outputting file links, save files to /mnt/data. [C4]

2. Available Resources

[C6]

Use the following predefined function(s):

search: {'query': {'description': 'Search query', 'type': 'str', 'required': 'True'}, 'recency_days': {'description': 'Recency of search results in days', 'type': 'int', 'required': 'False'}}

Example: Refer to the following example to solve the problem. Note that your response should follow the same format: [C7]

input: <task>I will use the search function to query "Beijing today temperature" to get today's temperature information for Beijing.</task>

output:

<code>

""" 1. Use the search function to query "Beijing today temperature" and retrieve relevant search results.

2. Print the search results for subsequent analysis. """

search_result = search(query="Beijing today temperature")

print(search_result)

</code>

[C8]

Query: <task>I need the latest information on Taylor Swift, including her profession, achievements, recent activities, related images, and videos. To obtain this information, I will use the search function to perform a web search and extract relevant details.</task>

Constraint Presentation Type

Vanilla Constraint

[C1] [C2] [C3] [C4]
[C5] [C8]

Condition Constraint

[C6]

Example Constraint

[C7]

Constraint Type

Semantic

Formatting

Tool

AGENTIF: the first benchmark for evaluating LLM instruction following in agentic scenarios

- (1) Realistic, constructed from **50 real-world** agentic applications.
- (2) Long, averaging **1723 words** with a maximum of 15630 words.
- (3) Complex, averaging **11.9 constraints per instruction**, covering diverse constraint types, such as tool specifications and condition constraints.

Benchmark	#Inst.	Len.	#Cons.	Data Resource	Constraint Type			Evaluation Method	
					Tool	Conditional	Example	Code-based	LLM-based
IFEval [37]	541	36	1.5	Synthetic	✗	✗	✗	✓	✗
FollowBench [14]	820	253	3.0	Synthetic	✗	✗	✓	✓	✓
InfoBench [23]	500	38	4.5	Synthetic	✗	✗	✗	✗	✓
SysBench [22]	500	521	2.4	Realistic	✗	✗	✗	✗	✓
ComplexBench [29]	1, 150	448	4.2	Synthetic	✗	✓	✗	✓	✓
AgentOrca [16]	663	1, 144	-	Synthetic	✓	✓	✗	✓	✗
Multi-IF [10]	4, 501	48	7.1	Synthetic	✗	✗	✗	✓	✗
AGENTIF (ours)	707	1, 723	11.9	Realistic	✓	✓	✓	✓	✓

Model Performance on AGENTIF

ISR: Instruction Success Rate
CSR: Constraint Success Rate

Models	Presentation Type			Type			ISR	CSR
	Vanilla	Condition	Example	Formatting	Semantic	Tool		
[T]GPT-5	61.6	33.7	80.0	64.7	61.7	45.6	30.2	60.8
[T]GLM-4.6	59.2	42.2	87.2	62.1	62.1	49.9	23.1	60.5
[N]Claude-4-Sonnet	59.1	44.3	83.6	62.2	62.2	45.5	21.5	60.1
[T]o1-mini	59.8	37.5	80.8	66.1	59.1	43.2	26.9	59.8
[N]Claude-3-7-Sonnet	60.9	38.9	69.2	60.1	61.3	50.9	23.3	59.5
[N]GPT-4o	58.0	35.1	80.8	65.8	56.5	43.2	26.4	58.5
[T]Qwen3-32B	57.5	41.1	80.6	57.7	62.5	45.7	24.9	58.4
[T]QwQ-32B	57.5	35.6	82.7	61.4	59.4	43.2	27.2	58.1
[T]DeepSeek-R1	56.1	41.4	87.0	61.4	58.9	44.4	22.2	57.9
[T]GLM-Z1-32B	56.7	37.9	83.6	60.2	59.6	43.1	23.8	57.8
[N]DeepSeek-V3	54.9	41.5	84.5	59.3	58.9	40.8	21.9	56.7
[N]Claude-3-5-Sonnet	57.3	36.9	69.2	61.5	56.0	43.3	24.9	56.6
[N]Meta-Llama-3.1-70B-Instruct	55.1	35.0	84.3	61.6	55.6	42.8	20.9	56.3
[T]DeepSeek-R1-Distill-Qwen-32B	54.5	39.6	73.1	55.7	57.2	45.2	20.7	55.1
[T]DeepSeek-R1-Distill-Llama-70B	55.4	37.7	69.2	56.5	56.6	44.1	19.9	55.0
[N]Meta-Llama-3.1-8B-Instruct	53.5	36.6	71.4	55.6	54.8	43.5	19.9	53.6
[S]Crab-DPO-7B	48.3	24.3	57.5	48.8	47.4	41.9	10.1	47.2
[N]Mistral-7B-Instruct-v0.3	47.9	29.2	53.8	47.0	48.6	39.8	11.5	46.8
[S]Conifer-DPO-7B	45.6	27.0	50.5	42.0	46.9	41.8	10.7	44.3

All models
demonstrate
suboptimal
performance

[N] denotes non-thinking models

[T] denotes thinking models

[S] denotes models explicitly designed for instruction following by the academic community

Model Performance on AGENTIF

Models	Presentation Type			Type			ISR	CSR
	Vanilla	Condition	Example	Formatting	Semantic	Tool		
[T]GPT-5	61.6	33.7	80.0	64.7	61.7	45.6	30.2	60.8
[T]GLM-4.6	59.2	42.2	87.2	62.1	62.1	49.9	23.1	60.5
[N]Claude-4-Sonnet	59.1	44.3	83.6	62.2	62.2	45.5	21.5	60.1
[T]o1-mini	59.8	37.5	80.8	66.1	59.1	43.2	26.9	59.8
[N]Claude-3-7-Sonnet	60.9	38.9	69.2	60.1	61.3	50.9	23.3	59.5
[N]GPT-4o	58.0	35.1	80.8	65.8	56.5	43.2	26.4	58.5
[T]Qwen3-32B	57.5	41.1	80.6	57.7	62.5	45.7	24.9	58.4
[T]QwQ-32B	57.5	35.6	82.7	61.4	59.4	43.2	27.2	58.1
[T]DeepSeek-R1	56.1	41.4	87.0	61.4	58.9	44.4	22.2	57.9
[T]GLM-Z1-32B	56.7	37.9	83.6	60.2	59.6	43.1	23.8	57.8
[N]DeepSeek-V3	54.9	41.5	84.5	59.3	58.9	40.8	21.9	56.7
[N]Claude-3-5-Sonnet	57.3	36.9	69.2	61.5	56.0	43.3	24.9	56.6
[N]Meta-Llama-3.1-70B-Instruct	55.1	35.0	84.3	61.6	55.6	42.8	20.9	56.3
[T]DeepSeek-R1-Distill-Qwen-32B	54.5	39.6	73.1	55.7	57.2	45.2	20.7	55.1
[T]DeepSeek-R1-Distill-Llama-70B	55.4	37.7	69.2	56.5	56.6	44.1	19.9	55.0
[N]Meta-Llama-3.1-8B-Instruct	53.5	36.6	71.4	55.6	54.8	43.5	19.9	53.6
[S]Crab-DPO-7B	48.3	24.3	57.5	48.8	47.4	41.9	10.1	47.2
[N]Mistral-7B-Instruct-v0.3	47.9	29.2	53.8	47.0	48.6	39.8	11.5	46.8
[S]Conifer-DPO-7B	45.6	27.0	50.5	42.0	46.9	41.8	10.7	44.3

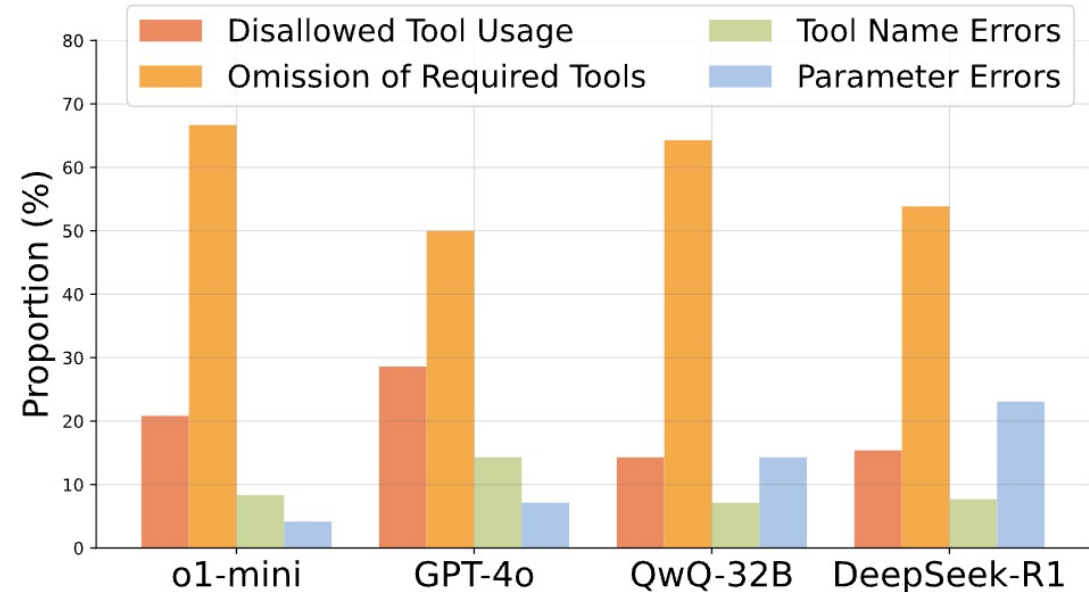
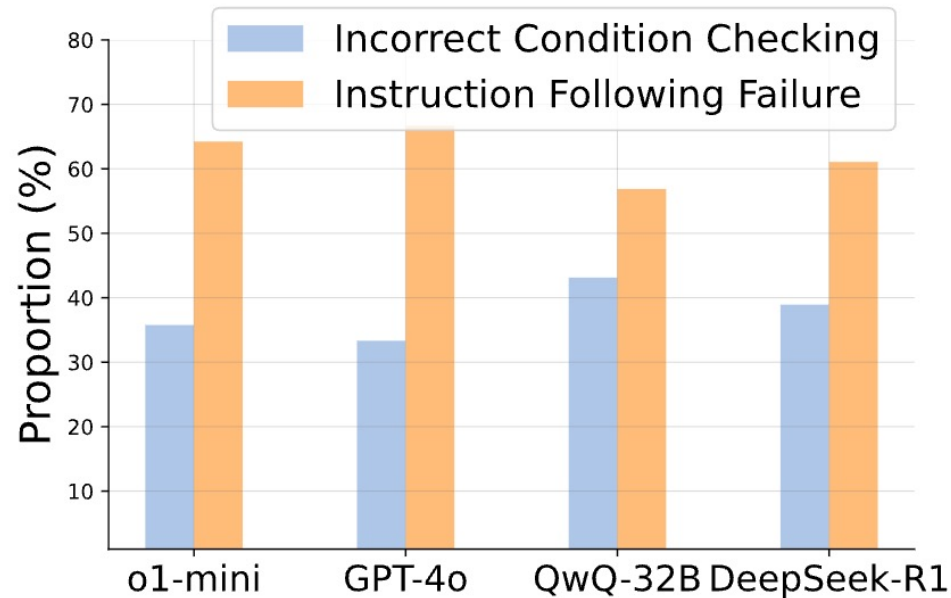
Models perform much lower on the **condition** and **tool** constraint

[N] denotes non-thinking models

[T] denotes thinking models

[S] denotes models explicitly designed for instruction following by the academic community

Analysis of Condition and Tool Constraint



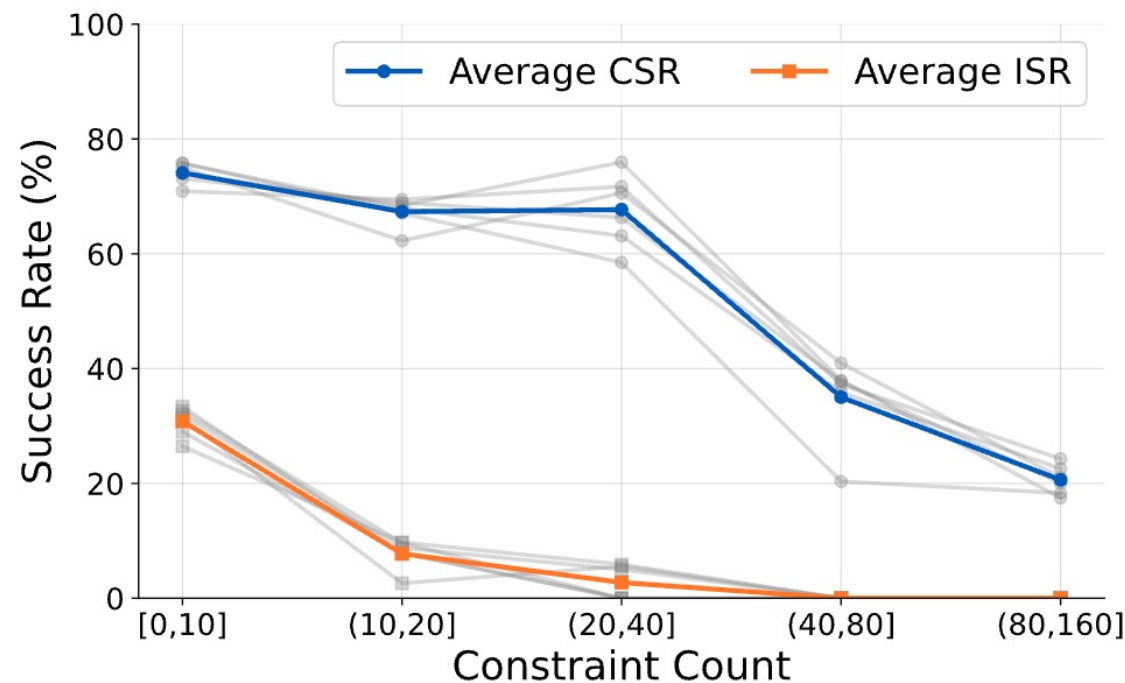
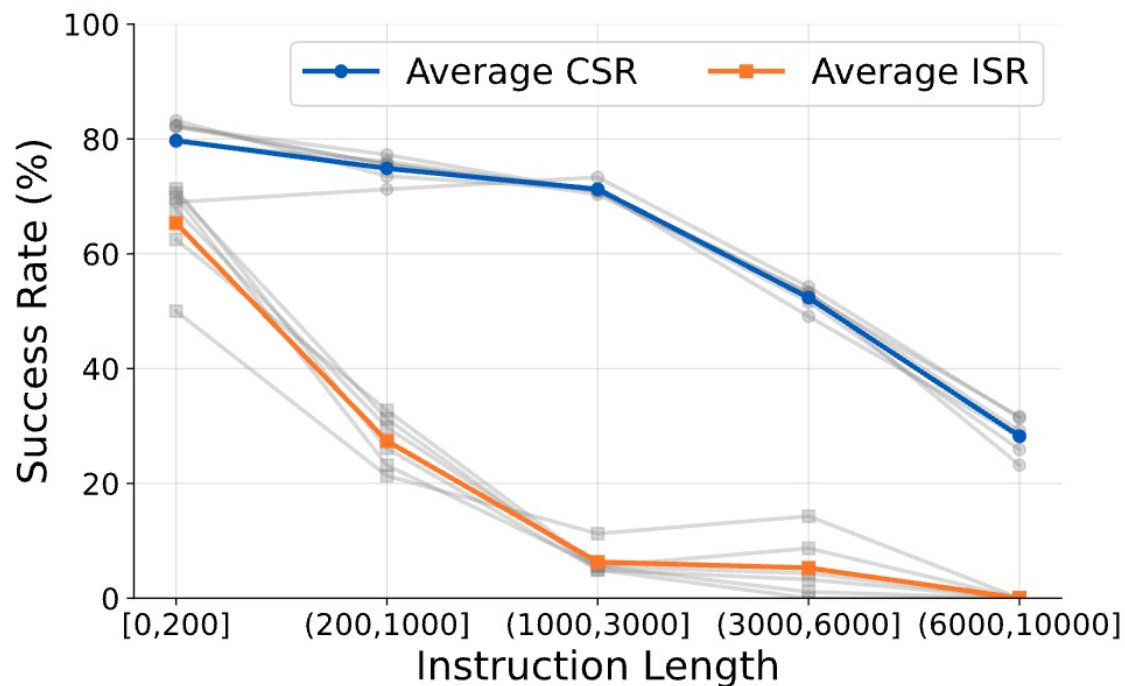
Condition Constraint:

A substantial portion (above 30%) of errors is due to incorrect condition checks

Tool Constraint:

- (1) Disallowed tool usage and omission of required tools constitute the primary errors.
- (2) Thinking models more frequently neglect the required tools.

Analysis of Instruction Length and Constraint Quantity



- Model performance generally declines with increasing instruction length or constraint count.
- When the sequence length exceeds 3,000 and the number of constraints is greater than 40, the model's performance exhibits a sharp decline, which may stem from increased task difficulty as well as insufficient relevant training data.

Analysis of Meta Constraints

Meta Constraint

Constraint Selection 91.4%

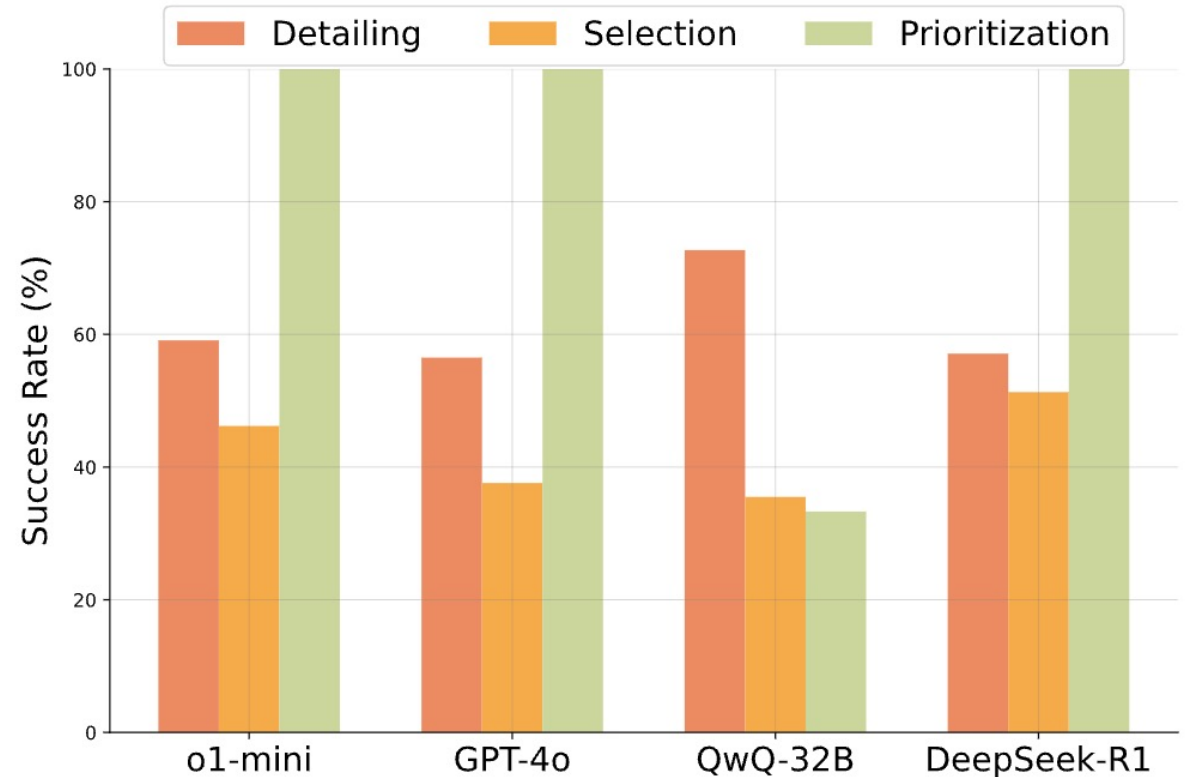

When a task lists **several constraints**, you are required to **satisfy only** the first constraint and may ignore the others.

Constraint Detailing 7.5%

You should address **both** “analyze emotional state” and “identify patterns” in separate sentences, using non-overlapping language and observations to avoid redundancy.

Constraint Prioritization 1.0%

If there is any **conflict between** adhering to the time-boxed agenda and covering all suggested talking points or discussion topics, **prioritize** maintaining the time-boxed structure.



Unlike regular constraints that apply directly to the model’s response, meta constraints govern other constraints

Conclusion

- We propose AgentIF, a benchmark for evaluating instruction following in realistic agentic scenarios with long, detailed instructions and complex constraints. We believe AgentIF is valuable to the community for building robust agentic applications.
- We conduct comprehensive evaluations using AgentIF and draw several insights. Our results show that even the best-performing LLM follows fewer than 30% of instructions perfectly, demonstrating the difficulty and necessity of AgentIF.
- We conduct further analyses, including error analysis, instruction length, and meta instructions, which reveal specific limitations of existing LLMs and potential directions for future optimization.



AgentIF



Thanks for Listening!

Contact: qyj23@mails.tsinghua.edu.cn

