GraphIF: Enhancing Multi-Turn Instruction Following for Large Language Models with Relation Graph Prompt

Zhenhe Li[1], Can Lin[1], Ling Zheng[1], Wen-Da Wei[2], Junli Liang[1], Qi Song[1]*

[1]University of Science and Technology of China
[2]School of Artificial Intelligence, Nanjing University, China
{lizh2002, can.lin, lingzheng, jlliang}@mail.ustc.edu.cn, weiwd@lamda.nju.edu.cn, qisong09@ustc.edu.cn

Abstract

Multi-turn instruction following is essential for building intelligent conversational systems that can consistently adhere to instructions across dialogue turns. However, existing approaches to enhancing multi-turn instruction following primarily rely on collecting or generating large-scale multi-turn dialogue datasets to fine-tune large language models (LLMs), which treat each response generation as an isolated task and fail to explicitly incorporate multi-turn instruction following into the optimization objectives. As a result, instruction-tuned LLMs often struggle with complex long-distance constraints. In multi-turn dialogues, relational constraints across turns can be naturally modeled as labeled directed edges, making graph structures particularly suitable for modeling multi-turn instruction following. Despite this potential, leveraging graph structures to enhance the multi-turn instruction following capabilities of LLMs remains unexplored. To bridge this gap, we propose GraphIF, a plug-and-play framework that models multi-turn dialogues as directed relation graphs and leverages graph prompts to enhance the instruction following capabilities of LLMs. GraphIF comprises three key components: (1) an agent-based relation extraction module that captures inter-turn semantic relations via action-triggered mechanisms to construct structured graphs; (2) a relation graph prompt generation module that converts structured graph information into natural language prompts; and (3) a response rewriting module that refines initial LLM outputs using the generated graph prompts. Extensive experiments on two long multi-turn dialogue datasets demonstrate that GraphIF can be seamlessly integrated into instruction-tuned LLMs and leads to significant improvements across all four multi-turn instruction-following evaluation metrics.

Code — https://github.com/sstillzh/GraphIF
Extended version — https://arxiv.org/abs/2511.10051

Introduction

Large Language Models (LLMs) (Achiam et al. 2023; Dubey et al. 2024; Team 2024) have demonstrated exceptional performance in dialogue systems. As conversational AI becomes increasingly important, the ability to understand and follow user instructions is crucial for effective interaction (Chiang et al. 2023; Zheng et al. 2023). Real-world conversations pose a challenge as users typically distribute requirements across multiple dialogue turns, making multi-turn instruction following essential for maintaining dialogue coherence and consistency.

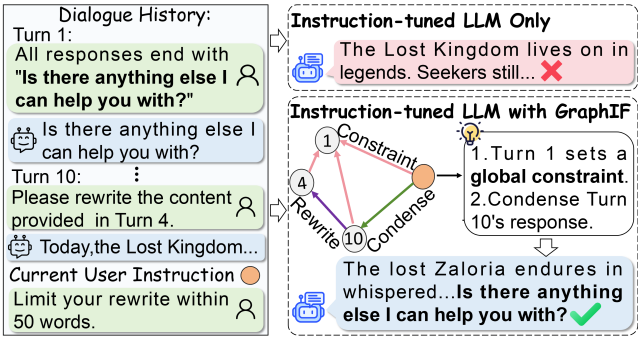

Figure 1: A comparison of two types of methods: Instruction-tuned LLM only and our proposed GraphIF that uses graph structure to enhance the multi-turn instruction following.

Recent efforts to enhance multi-turn instruction following have primarily focused on collecting (Chiang et al. 2023; Zhao et al. 2024) or generating (Maosongcao et al. 2025; Gao et al. 2025) multi-turn dialogue datasets to fine-tune LLMs. However, even well-tuned LLMs struggle with complex long-distance constraints (Kwan et al. 2024). As illustrated in Figure 1, LLMs often forget instructions established in earlier turns, such as ending responses with specific phrases. This limitation stems from the mismatch between autoregressive training and the non-sequential dependencies in multi-turn dialogues—current instruction-tuning methods treat each response generation as an isolated task without explicitly modeling cross-turn constraints (Zhang et al. 2025).

While incremental training could mitigate this issue, it incurs substantial computational costs and suffers from weak generalization due to data limitations (Li et al. 2025b). This raises a fundamental question: can we develop training-free approaches that enable LLMs to satisfy inter-turn relational constraints without parameter updates?

In order to satisfy inter-turn relational constraints, the

*Corresponding author

LLM should have the ability to remember the previous dialogues, correctly identify the related contexts, and then accordingly generate answers to the instruction in the current turn. This imposes two challenges: 1) **How to extract the semantic relations between different turns?** As illustrated in Figure 1, turn 1 introduces a global constraint requiring all responses to end with a specific sentence, and the current instruction requests compression of turn 10's response. Extracting such inter-turn relations remains non-trivial. 2) **How to leverage the extracted relations to generate correct responses?** Once the model identifies the specific relations linking the current instruction to turns 1 and 10, how can these captured relations be effectively utilized to generate responses that satisfy all relevant constraints?

Graph structure has been proven to be able to represent the semantic relations in scenarios including multi-document QA (Wang et al. 2024b; Edge et al. 2024) and long-context QA (Li et al. 2024). In multi-turn dialogues, the relational constraints between different dialogue turns can be naturally represented through labeled directed edges, making graph structures particularly suitable for modeling multi-turn instruction following (Li et al. 2025a). Despite existing work on using graph structures to generate multi-turn dialogue datasets (Li et al. 2025a), leveraging graph structures to enhance multi-turn instruction following remains unexplored.

To address the aforementioned challenges and bridge the gap, we propose GraphIF, a training-free and plug-and-play framework that models multi-turn dialogues as directed relation graphs and leverages graph prompts to enhance multi-turn instruction following for LLMs. We use graph structures to uniformly model inter-turn relations as structured information. As shown in Figure 1, each node in the relation graph represents a dialogue turn, and labeled directed edges capture constraint relations between connected turns. The directed edge `<Current Turn, Global Constraint Imposed By, Turn 1>` indicates that Turn 1 imposes a global constraint that the current response must satisfy. Key components of GraphIF are as follows:

(1) We design an agent-based relation extraction module. Directly extracting semantic relations between dialogue turns is difficult, especially for complex long-distance constraints (Bai et al. 2024; Sun et al. 2024). Inspired by agent-based decomposition paradigms (Yao et al. 2023; Erdogan et al. 2025), we design an agent-based relation extraction module that employs LLM to alternate between action identification and action execution phases to iteratively construct the dialogue relation graph.

(2) We design two modules to leverage the extracted relations for generating correct responses. We first design a relation graph prompt generation module that converts the structured graph information into natural language prompts, explicitly articulating inter-turn relations and their corresponding constraints. Finally, we design an initial response rewrite module that leverages the constraints articulated in the graph prompts to refine the initial responses generated by LLMs. This refinement ensures better adherence to multi-turn dialogue constraints.

Given that existing multi-turn instruction following datasets have limited dialogue turns and oversimplified inter-turn relations (Zheng et al. 2023; He et al. 2024; Bai et al. 2024), we construct two new datasets with extended dialogues and complex semantic relations based on two prior benchmarks (Kwan et al. 2024; Li et al. 2025a). Comprehensive experiments on the two datasets demonstrate that current fine-tuned LLMs exhibit suboptimal performance when confronted with complex constraints. Our GraphIF framework can be seamlessly integrated into existing LLMs and achieves significant improvements across all four multi-turn instruction-following evaluation metrics.

We summarize our contributions as follows:

- We propose GraphIF, a training-free and plug-and-play framework that explicitly models inter-turn relations through graph structures and generates graph prompts to refine initial LLM responses, thereby enhancing multi-turn instruction following capabilities of LLMs.
- We design an agent-based relation extraction module that iteratively performs action identification and execution to construct dialogue relation graphs, addressing the challenge of difficult relation extraction. To leverage the extracted relations for generating correct responses, we develop a relation graph prompt generation module that converts graph structures into natural language prompts, and a response rewrite module that uses graph prompts to refine LLM outputs.
- Extensive experiments on two long multi-turn dialogue datasets demonstrate that GraphIF can be seamlessly integrated into existing instruction-tuned LLMs and leads to significant improvements across all four multi-turn instruction-following evaluation metrics.

Related Work

Instruction Fine-tuning for LLMs. Current approaches for enhancing multi-turn instruction following mainly use supervised fine-tuning with instruction datasets. Some methods curate real-world user-LLM interactions (Wang et al. 2024a; Zhao et al. 2024), while others generate synthetic dialogues (Ding et al. 2023; Wu et al. 2025; Chen et al. 2025). Parrot (Sun et al. 2024) addresses anaphora and ellipsis by training specialized models. However, these instruction-tuning methods overlook explicit modeling of inter-turn relational structures.

Graph-Augmented Generation with LLMs. Recent studies explore integrating graph structures to enhance LLM capabilities (Jimenez Gutierrez et al. 2024; Li et al. 2024; Lin et al. 2025). Specifically, GraphRAG (Edge et al. 2024), LightRAG (Guo et al. 2024), and KGP (Wang et al. 2024b) utilize cross-document or passage-level graphs to improve retrieval and information aggregation. Despite these advances, leveraging graph structures to enhance multi-turn instruction following remains unexplored.

Preliminary

Multi-Turn Instruction Following We formally define the *multi-turn instruction following task* as $\mathcal{D} = \{(\mathcal{H}_t, \mathcal{I}_t)\}_{t=1}^{M}$, where M represents total number of dialogue turns ($M > 1$), $\mathcal{I}_t$ represents the user instruction in t-th dialogue turn,

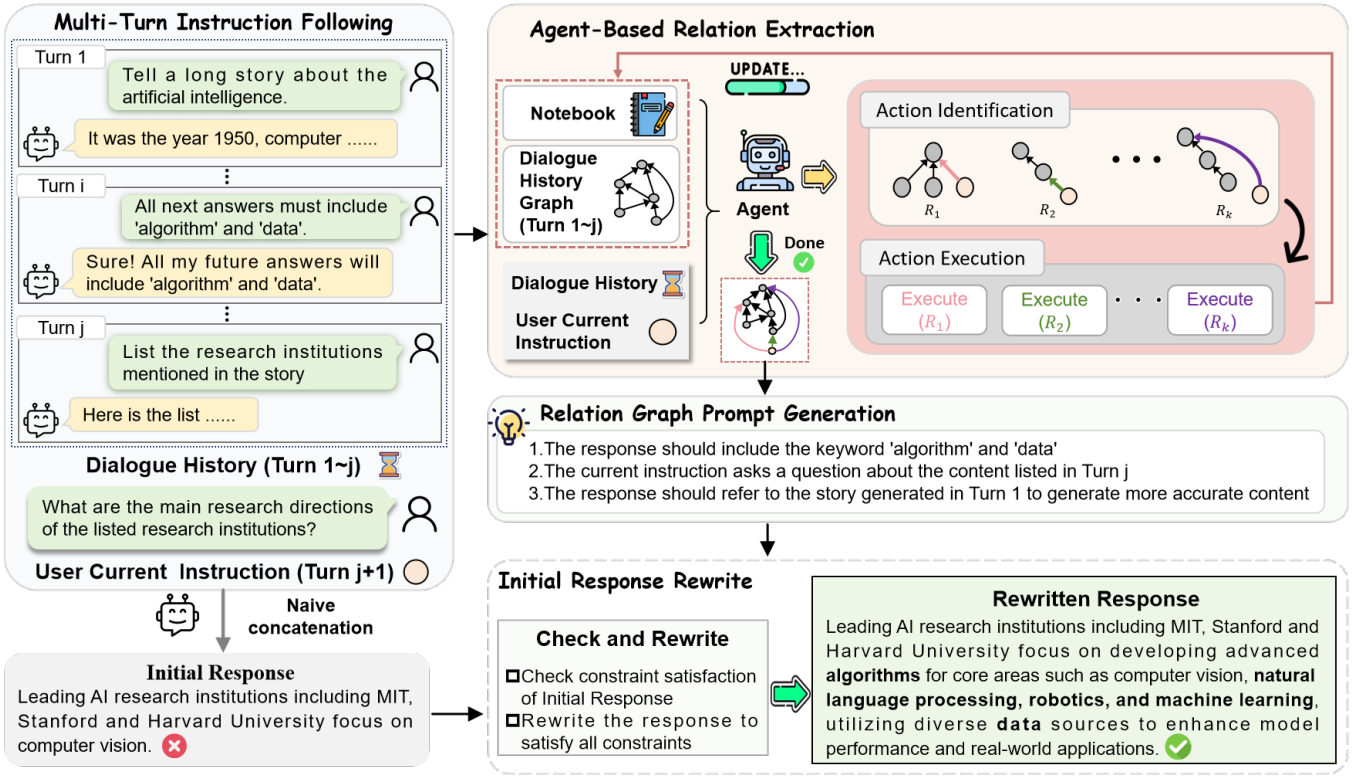

Figure 2: Framework overview of GraphIF. Given dialogue history and current user instruction, GraphIF first extracts semantic relations between dialogues through the *Agent-Based Relation Extraction* module, then employs *Relation Graph Prompt Generation* to generate constraint-aware prompts, and finally uses the *Initial Response Rewrite* module to refine the initial response.

and $\mathcal{H}_t = \{\langle \mathcal{I}_k, RES_k \rangle\}_{k=1}^{t-1}$ represents the dialogue history before turn t, where RES_k means the response generated by LLM in k-th dialogue turn. Specifically, $\mathcal{H}_1$ is empty in the first dialogue turn, and the LLM directly generates a response to $\mathcal{I}_1$. For definitional rigor, our subsequent discussion focuses on scenarios where $t \geq 2$, ensuring that the dialogue history $\mathcal{H}_t$ is non-empty. Given an instruction $\mathcal{I}_t$ and dialogue history $\mathcal{H}_t$, the goal is to generate an overall response RES_t that aligns with the given context while maintaining semantic consistency across turns.

Relation Graph in GraphIF We employ the directed relation graph to model the multi-turn instruction-following dialogue scenario. In the graph, each vertex represents a dialogue turn, while the directed edges between vertices capture the semantic relations across dialogue turns. Formally, let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{R}\}$ be the constructed relation graph, where each node $v_i \in \mathcal{V}$ represents a complete dialogue turn $< \mathcal{I}_i, RES_i >$. And $\mathcal{E} \subset \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ represents the edge set, where $\mathcal{R}$ denotes the predefined relation labels between dialogue turns.

Methodology

In this section, we introduce the technical details of GraphIF. Figure 2 illustrates the overall framework. GraphIF mainly contains three components: (1) Agent-Based Relation Extraction module, which extracts relations between dialogue turns to construct the structured graph, (2) Relation Graph Prompt Generation module, which transforms the structured graph information into natural language prompts, (3) Initial Response Rewrite module, which leverages the graph prompt to refine the initial response generated by LLMs.

Agent-Based Relation Extraction

In multi-turn instruction following scenarios, complex contextual semantic relations exist between current user instructions and dialogue history. To model these interactions, we explicitly identify and extract semantic relations between current instructions and historical dialogue turns, constructing a semantic relation graph. Formally, given user instruction $\mathcal{I}_t$ and dialogue history $\mathcal{H}_t$ at turn t, we identify specific semantic relations between $\mathcal{I}_t$ and targeted turns in $\mathcal{H}_t$.

Given complex relational dependencies between current instructions and multiple dialogue turns, single-step relation extraction is insufficient. Inspired by agent-based problem decomposition paradigms (Yao et al. 2023; Erdogan et al. 2025), we propose an iterative agent-based module for progressive relation extraction. Since different dialogue relations require tailored identification approaches, our module decomposes relation extraction into two complementary subtasks: **Action Identification** and **Action Execution**.

In Action Identification, we introduce an action-triggered mechanism to identify corresponding relations. Formally, given the predefined dialogue relation set $\mathcal{R} =$

$\{r_1, r_2, \ldots, r_m\}$, each relation label $r_i \in \mathcal{R}$ corresponds to an action $a_i \in \mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ that uniquely identifies it. During the iterative relation extraction process, $a_{ts'}$ represents the action identified at timestep s' of turn t, indicating the identification of the corresponding relation $r_{ts'}$. The iterative process continues until the termination action Done is identified, indicating that all relevant relations have been extracted.

In Action Execution, each action a_i maintains a dedicated implementation function $Execute(a_i)$ to extract dialogue turns satisfying the identified relations with the current instruction. For the identified action $a_{ts'}$, we typically denote the execution result as $E_{ts'}$, which represents the set of historical dialogue turns that satisfy the relation $r_{ts'}$ with the current instruction. These identified turns define directed edges in the relation graph, where each edge points from the current instruction to a historical dialogue turn, representing the relation $r_{ts'}$ between them.

Additionally, to enable incremental updates while avoiding redundant extraction, we introduce a dynamic notebook mechanism $\mathcal{N} = \{a_{ts'} \rightarrow E_{ts'}\}$ that maintains relation mappings across iterations, allowing the agent to progressively construct the dialogue relation graph.

In the following, we detail the concrete implementation of the action identification and action execution phases.

Action Identification Given the potential existence of multiple relations between $\mathcal{I}_t$ and $\mathcal{H}_t$, the agent employs an iterative updating mechanism, where the action identification module identifies the most critical relation at each timestep and updates the notebook accordingly. Specifically, at timestep s, based on the extracted information maintained in the notebook $\mathcal{N}$, we prompt the LLM to identify the next relation instance from the dialogue history and return the corresponding action a_{ts} (return Done action if all required content has been extracted):

$$a_{ts} = LLM(\mathcal{I}_t, \mathcal{H}_t, \mathcal{N}). \tag{1}$$

Action Execution Upon action identification, the identified action a_{ts} corresponds to relation $r_{ts} \in \mathcal{R}$, and the corresponding implementation function $Execute(a_{ts})$ will be executed. Analogous to the action identification phase, we leverage the notebook $\mathcal{N}$ to maintain the memory of extracted relations, thereby eliminating redundant relation recognition. Typically, we prompt LLM to locate the E_{ts}:

$$E_{ts} = LLM(\mathcal{I}_t, \mathcal{H}_t, r_{ts}, \mathcal{N}). \tag{2}$$

After finishing locating E_{ts}, we update the notebook $\mathcal{N}$ and dialogue relation graph $\mathcal{G}$:

$$\mathcal{N} = \mathcal{N} \cup (a_{ts} \rightarrow E_{ts}), \tag{3}$$

$$\mathcal{G} = \mathcal{G} + \bigcup_{i \in E_{ts}} (v_t, r_{ts}, v_i). \tag{4}$$

Here, we update the notebook with the identified action a_{ts} and the relevant dialogue turns E_{ts}. Additionally, each dialogue turn is connected to the current user instruction by an edge of relation type r_{ts}.

Implementation We define a set of actions that characterize representative dialogue relations. Unlike previous benchmarks (Kwan et al. 2024; Li et al. 2025a), we define a more inclusive and coarse-grained set of relations that better capture the dialogue constraints in real-world scenarios.

Identify_Global_Constraint: Identifies the current instruction as a global constraint for subsequent interactions, adding it to the global constraint set.

Identify_Context_Anchored: Identifies specific historical dialogue content that the instruction semantically relies on or logically connects to. The implementation locates the most relevant unrecorded dialogue turn.

Identify_Modify: Identifies specific historical dialogue content that the current instruction refines or extends, built upon context-anchored relations with additional requirements. The implementation follows the same approach as context-anchored relation extraction.

Identify_Summary: Identifies the instruction that requests a summary of specific historical dialogue turns. The implementation locates all dialogue turns to be summarized.

New_Topic: Identifies instructions that introduce entirely new topics unconnected to the current dialogue context. The implementation determines whether the topic shifts back to a previous topic or initiates a new topic.

Done: Indicates complete relation extraction termination.

Relation Graph Prompt Generation

We convert the extracted dialogue relations into interpretable prompt formats to enhance contextual understanding of LLMs.

Concretely, we construct the graph prompt $\mathcal{P}_g$ through a structured concatenation process. For each identified semantic relation r_{ts}, we systematically integrate three parts: (1) the formal definition of r_{ts}, (2) the specific description of how this relation connects to the current instruction $\mathcal{I}_t$, i.e., the specific constraints that the response should satisfy and (3) dialogue content of the corresponding turns.

Figure 2 illustrates the core content of $\mathcal{P}_g$, including the identified semantic relations and the constraints that the response should satisfy.

Initial Response Rewrite

We generate the initial response by directly concatenating the dialogue history with the user instruction, which represents the implementation of LLM in real-world applications:

$$RES_{initial} = LLM([\mathcal{H}_t, \mathcal{I}_t]). \tag{5}$$

As illustrated in Figure 2, $RES_{initial}$ fails to incorporate the two specified keywords and exhibits inaccuracies, omitting crucial information from the original story.

Then we leverage the constructed graph prompt to refine the initial response:

$$RES_t = \text{Rewrite}(RES_{initial}, \mathcal{P}_g). \tag{6}$$

As illustrated in Figure 2, the LLM identifies unsatisfied constraints in the $RES_{initial}$ by examining the content of $\mathcal{P}_g$, and subsequently performs targeted rewriting to generate a refined response RES_t that incorporates the two required keywords and addresses the key information missing from the $RES_{initial}$.

Backbone Model	Method	MT-Eval*(%)				StructFlowBench*(%)			
		CSR	ISR	DRFR	WCSR	CSR	ISR	DRFR	WCSR
Qwen2.5-7B-Instruct	LLM-only	80.22	51.30	79.70	79.57	70.70	23.02	71.22	67.07
	+MemoryBank	70.29	30.87	69.84	67.88	74.63	29.99	74.92	70.58
	+MemoChat	59.42	16.52	58.79	56.35	72.87	27.08	73.23	68.72
	+GraphIF	**91.30**	**76.96**	**91.06**	**90.72**	**89.46**	**69.25**	**89.47**	**88.60**
Llama-3.1-8B-Instruct	LLM-only	67.03	27.51	67.88	64.22	71.03	29.32	72.75	67.98
	+MemoryBank	67.17	31.74	66.97	67.74	74.37	35.85	75.48	70.83
	+MemoChat	56.30	15.22	55.45	53.97	73.51	32.07	74.29	69.86
	+GraphIF	**91.27**	**80.35**	**90.87**	**92.58**	**88.91**	**70.29**	**89.38**	**88.04**
Hermes-3-Llama-3.1-8B	LLM-only	76.74	48.70	76.21	76.09	68.39	23.55	69.74	65.11
	+MemoryBank	68.41	34.35	68.33	69.07	69.20	23.57	70.03	64.98
	+MemoChat	54.93	11.30	53.94	52.26	68.79	24.64	69.89	64.76
	+GraphIF	**86.96**	**70.43**	**86.52**	**86.64**	**78.74**	**46.74**	**79.73**	**76.54**
Llama-3.1-Storm-8B	LLM-only	80.36	56.96	81.36	80.87	74.07	34.12	74.99	70.75
	+MemoryBank	66.16	28.70	65.45	66.35	73.92	31.94	74.37	70.02
	+MemoChat	55.36	13.04	54.39	53.59	71.55	30.60	71.62	67.78
	+GraphIF	**93.62**	**81.30**	**93.33**	**94.87**	**86.92**	**64.18**	**86.83**	**85.32**
Llama-3.1-Tulu-3.1-8B	LLM-only	74.13	44.35	75.30	72.70	80.92	47.85	81.93	78.78
	+MemoryBank	73.70	43.04	73.03	73.45	78.51	42.24	78.88	75.32
	+MemoChat	50.65	19.57	51.36	49.83	64.71	25.13	66.46	61.85
	+GraphIF	**88.48**	**76.96**	**88.03**	**89.68**	**88.68**	**69.71**	**88.98**	**87.90**

Table 1: Performance comparison of different methods across five backbone LLMs on MT-Eval* and StructFlowBench* datasets. LLM-only: direct concatenation of dialogue history and user instruction for response generation; Others: integration of respective methods into LLM. Best results are highlighted in bold.

Model	Method	MT-Eval*(%)			
		CSR	ISR	DRFR	WCSR
Qwen2.5-3B -Instruct	LLM-Only	59.42	9.56	59.09	56.99
	+GraphIF	**75.43**	**46.09**	**75.45**	**75.51**
Qwen2.5-7B -Instruct	LLM-Only	80.22	51.30	79.70	79.57
	+GraphIF	**91.30**	**76.96**	**91.06**	**90.72**
Qwen2.5-14B -Instruct	LLM-Only	90.22	75.65	90.00	92.61
	+GraphIF	**93.62**	**82.17**	**93.48**	**95.71**

Table 2: Performance comparison of Qwen2.5 models of different scales as backbone models on the MT-Eval*.

Experiments

Experimental Settings

Dataset Existing multi-turn instruction-following datasets are limited by short dialogue turns (typically under eight turns) (Bai et al. 2024; He et al. 2024), inadequately representing complex multi-turn instruction following scenarios. To address this, we construct evaluation datasets following two benchmarks that model inter-turn relations. We design MT-Eval* by merging and manually verifying multiple instances from MT-Eval (Kwan et al. 2024). Similarly, we leverage the customizable structural framework of StructFlowBench (Li et al. 2025a) to design StructFlowBench*. The two datasets we construct feature dialogues with over 20 turns, each containing multiple inter-turn relations within individual dialogue instances.

Backbone Model We evaluate five instruction-tuned LLMs as backbone architectures, comprising two official models: Llama-3.1-8B-Instruct (Dubey et al. 2024) and Qwen2.5-7B-Instruct (Team 2024), along with three models that are fine-tuned and aligned through preference-based methods based on Llama-3.1-8B: Hermes-3-Llama-3.1-8B[1], Llama-3.1-Storm-8B[2], and Llama-3.1-Tulu-3.1-8B[3] (Lambert et al. 2024).

Baseline Due to the lack of robust frameworks specifically designed for multi-turn instruction-following enhancement, we select two representative memory-enhanced frameworks as baselines:

- MemoryBank (Zhong et al. 2024) summarizes dialogue content as events and employs RAG mechanisms to retrieve relevant conversations, enhancing models' memory of key information.
- MemoChat (Lu et al. 2023) creates topic-indexed storage structures and summarizes related dialogues, using LLM-based retrieval to enhance response generation.

Evaluation We adopt four instruction-following metrics to evaluate models from different perspectives: Constraint Satisfaction Rate (CSR), Instruction Satisfaction Rate (ISR) (Zhang et al. 2024), Decomposed Requirements Following Ratio (DRFR) (Qin et al. 2024), and Weighted Constraint

[1]https://huggingface.co/NousResearch/Hermes-3-Llama-3.1-8B

[2]https://huggingface.co/akjindal53244/Llama-3.1-Storm-8B

[3]https://huggingface.co/allenai/Llama-3.1-Tulu-3.1-8B

Dataset	Model	Method	Results(%) CSR	ISR	DRFR	WCSR
MT-Eval*	Qwen2.5-7B-Instruct	GraphIF	**91.30**	**76.96**	**91.06**	**90.72**
		w/o Relation Extraction Agent	67.17	21.74	66.52	63.48
		w/o Graph Prompt	85.80	63.48	85.45	84.14
	Llama-3.1-8B-Instruct	GraphIF	**91.27**	**80.35**	**90.87**	**92.58**
		w/o Relation Extraction Agent	73.77	40.87	73.33	72.72
		w/o Graph Prompt	84.93	63.32	84.47	85.07
StructFlowBench*	Qwen2.5-7B-Instruct	GraphIF	**89.46**	**69.25**	**89.47**	**88.60**
		w/o Relation Extraction Agent	73.36	27.71	74.11	69.53
		w/o Graph Prompt	75.93	30.95	75.74	72.34
	Llama-3.1-8B-Instruct	GraphIF	**88.91**	**70.29**	**89.38**	**88.04**
		w/o Relation Extraction Agent	81.74	55.08	81.99	79.61
		w/o Graph Prompt	77.56	44.11	77.84	74.82

Table 3: The results of ablation study. "w/o Relation Extraction Agent" refers to removing the agent module and directly extracting all inter-turn relations via one-time LLM inference, "w/o Graph Prompt" denotes using only relevant dialogue content without semantic relation explanations.

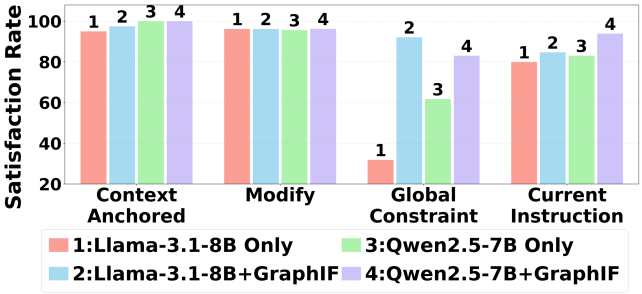

Figure 3: Detailed constraint satisfaction results across different constraint types on the MT-Eval*.

Satisfaction Rate (WCSR) (Li et al. 2025a). Following recent benchmarks (Zheng et al. 2023; Li et al. 2025a), we utilize GPT-4o as the LLM judge with human verification.

Implementation Details We set the temperature parameter of the LLMs to 0.7, top_p to 0.8, and top_k to 20 for achieving a balance between response diversity and generation stability. For evaluation, we conduct three independent experimental runs and compute the average of the evaluation metrics as the results. All experiments are conducted on 4 A800-80GB GPUs.

Main Results

Overall Performance Comparison. Table 1 presents the comprehensive evaluation of GraphIF against baseline methods on both MT-Eval* and StructFlowBench* datasets. Integrating GraphIF into instruction-tuned LLMs yields substantial improvements across all multi-turn instruction following metrics, with performance gains ranging from 7% to 52%. The most striking improvement appears in the Instruction Satisfaction Rate (ISR), the most stringent metric that scores only when responses satisfy all constraints within a dialogue turn. GraphIF achieves over 20% improvement in ISR, indicating a significant increase in the proportion of dialogues that perfectly adhere to all specified constraints.

In contrast, memory-enhanced approaches (MemoryBank and MemoChat) fail to improve over vanilla instruction-tuned LLMs, showing negligible gains or even hurting performance. This highlights the fundamental limitations of memory-based methods in capturing complex relational structures in multi-turn dialogues.

Limitations of Memory-Enhanced Mechanisms. All Memory-enhanced methods fail to significantly improve multi-turn instruction-following capabilities. MemoryBank employs coarse-grained summaries with naive RAG mechanisms that retrieve content based solely on vector similarity, missing crucial inter-turn semantic relations. MemoChat's fine-grained topic-based indexing fails to capture macro-level dialogue structure, often retrieving irrelevant information that interferes with contextual understanding.

Performance Analysis Across Constraint Types. Figure 3 reveals GraphIF's effectiveness across different constraint types on MT-Eval*. While Instruction-tuned LLMs struggle particularly with global constraints and show performance drops of up to 40%, GraphIF achieves remarkable improvements in this challenging constraint type. Beyond global constraints, GraphIF maintains or enhances performance across all constraint types, demonstrating its robust capability to adapt to diverse dialogue challenges. The improvement stems from GraphIF's ability to identify relevant relation types and locate specific dialogue turns that establish these relations. Results on StructFlowBench* demonstrate similar experimental results. Notably, GraphIF improves satisfaction of current constraints, simultaneously addressing immediate user requirements and maintaining historical consistency.

Scalability Across Model Sizes. Table 2 demonstrates GraphIF's effectiveness on MT-Eval* across LLMs of varying sizes. The consistent improvements across all model scales validate the generalizability and robustness of

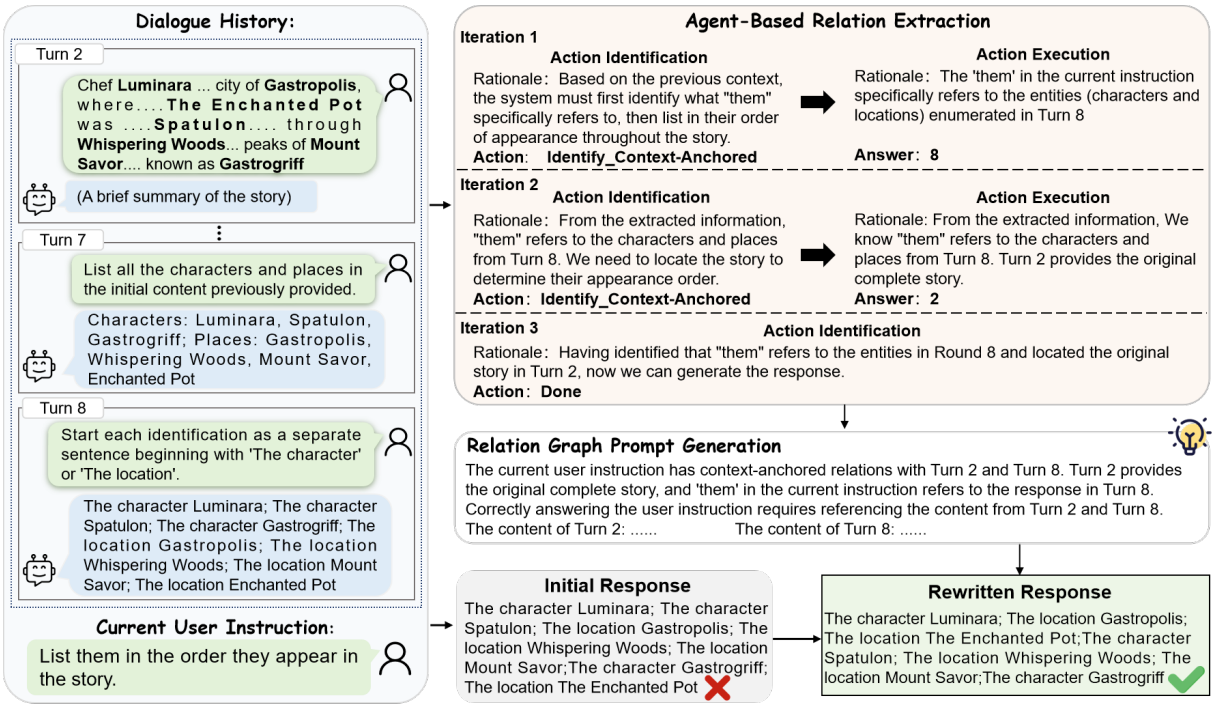

Figure 4: A typical case demonstrating how GraphIF iteratively extracts relations and corrects errors in the initial response through generated relation graph prompts.

GraphIF. Similar results are observed on StructFlowBench*. As a plug-and-play framework, GraphIF integrates seamlessly with any instruction-tuned LLM, enhancing multi-turn dialogue capabilities without requiring retraining or architectural modifications.

Ablation Study

The Effect of Agent-Based Relation Extraction. To evaluate the iterative agent-based module, we compare it against a one-time LLM inference baseline for relation extraction. Table 3 shows that direct extraction suffers a marked performance decline. This degradation suggests that a single-pass inference is insufficient for capturing complex relations within multi-turn dialogues, thereby underscoring the necessity of our iterative agent-based mechanism.

The Effect of Relation Graph Prompt Generation. To assess the Relation Graph Prompt Generation module, we conduct an ablation by replacing graph prompts with relation-free versions that contain only dialogue content without semantic relation explanations. As shown in Table 3, the significant performance drop underscores that LLMs struggle to autonomously infer inter-turn relations from raw dialogue. This highlights the necessity of explicit graph prompts in enabling models to navigate and satisfy complex inter-turn constraints.

Case Study

We present a specific case of Llama-3.1-8B-Instruct in Figure 4. The current user instruction is "List them in the order they appear in the story", which requires reordering entities from turn 8 according to their appearance sequence in the story from turn 2. The initial response of LLM exhibits ordering confusion issues. Through iterative relation extraction, GraphIF identifies the Context-Anchored relation between the current instruction and the dialogue content of turn 2 and turn 8, and generates a graph prompt based on this relation to explain the semantic connection. Guided by the graph prompt, the LLM adjusts the entity ordering in the initial response, thereby generating the correct response. This example demonstrates that GraphIF can rectify errors in initial responses through accurate relation identification, thus improving response satisfaction for current instructions.

Conclusion

We propose GraphIF, a training-free and plug-and-play framework that models multi-turn dialogues as directed relation graphs and leverages graph prompts to enhance the instruction-following capabilities of LLMs. We carefully design two long multi-turn dialogue datasets based on public benchmarks to better evaluate the instruction-following capabilities. Extensive experiments on both datasets demonstrate that GraphIF can be seamlessly integrated into instruction-tuned LLMs, achieving significant improvements across all four evaluation metrics and maintaining consistent performance gains across different model scales. These results suggest that GraphIF provides a practical solution for enhancing multi-turn instruction following for LLMs in real-world applications.

Acknowledgments

The research was partially supported by the China National Natural Science Foundation with no. 62132018, and Hefei Key Technology Research and Development Project (2024SZD005).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, G.; Liu, J.; Bu, X.; He, Y.; Liu, J.; Zhou, Z.; Lin, Z.; Su, W.; Ge, T.; Zheng, B.; et al. 2024. MT-Bench-101: A Fine-Grained Benchmark for Evaluating Large Language Models in Multi-Turn Dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7421–7454.

Chen, J.; Guan, X.; Yuan, Q.; Mo, G.; Zhou, W.; Lu, Y.; Lin, H.; He, B.; Sun, L.; and Han, X. 2025. ConsistentChat: Building Skeleton-Guided Consistent Dialogues for Large Language Models from Scratch. *arXiv preprint arXiv:2506.03558*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6.

Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Hu, S.; Liu, Z.; Sun, M.; and Zhou, B. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 3029–3051.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv e-prints*, arXiv–2407.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Erdogan, L. E.; Furuta, H.; Kim, S.; Lee, N.; Moon, S.; Anumanchipalli, G.; Keutzer, K.; and Gholami, A. 2025. Plan-and-Act: Improving Planning of Agents for Long-Horizon Tasks. In *Forty-second International Conference on Machine Learning*.

Gao, X.; Pei, Q.; Tang, Z.; Li, Y.; Lin, H.; Wu, J.; Wu, L.; and He, C. 2025. A Strategic Coordination Framework of Small LLMs Matches Large LLMs in Data Synthesis. *arXiv preprint arXiv:2504.12322*.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779*.

He, Y.; Jin, D.; Wang, C.; Bi, C.; Mandyam, K.; Zhang, H.; Zhu, C.; Li, N.; Xu, T.; Lv, H.; et al. 2024. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*.

Jimenez Gutierrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37: 59532–59569.

Kwan, W.-C.; Zeng, X.; Jiang, Y.; Wang, Y.; Li, L.; Shang, L.; Jiang, X.; Liu, Q.; and Wong, K.-F. 2024. MT-Eval: A Multi-Turn Capabilities Evaluation Benchmark for Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 20153–20177.

Lambert, N.; Morrison, J.; Pyatkin, V.; Huang, S.; Ivison, H.; Brahman, F.; Miranda, L. J. V.; Liu, A.; Dziri, N.; Lyu, S.; et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Li, J.; Li, J.; Wang, Y.; Chang, Y.; and Wu, Y. 2025a. StructFlowBench: A Structured Flow Benchmark for Multi-turn Instruction Following. In *Findings of the Association for Computational Linguistics: ACL 2025*, 9322–9341.

Li, S.; He, Y.; Guo, H.; Bu, X.; Bai, G.; Liu, J.; Liu, J.; Qu, X.; Li, Y.; Ouyang, W.; et al. 2024. GraphReader: Building Graph-based Agent to Enhance Long-Context Abilities of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 12758–12786.

Li, Y.; Shen, X.; Yao, X.; Ding, X.; Miao, Y.; Krishnan, R.; and Padman, R. 2025b. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*.

Lin, C.; Jiang, Z.; Zheng, L.; Zhao, Q.; Zhang, Y.; Song, Q.; and Zhou, W. 2025. RJE: A Retrieval-Judgment-Exploration Framework for Efficient Knowledge Graph Question Answering with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 17288–17305.

Lu, J.; An, S.; Lin, M.; Pergola, G.; He, Y.; Yin, D.; Sun, X.; and Wu, Y. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Maosongcao, M.; Zhang, T.; Li, M.; Zhang, C.; Liu, Y.; He, C.; Duan, H.; Zhang, S.; and Chen, K. 2025. Condor: Enhance llm alignment with knowledge-driven data synthesis and refinement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 22392–22412.

Qin, Y.; Song, K.; Hu, Y.; Yao, W.; Cho, S.; Wang, X.; Wu, X.; Liu, F.; Liu, P.; and Yu, D. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Sun, Y.; Liu, C.; Zhou, K.; Huang, J.; Song, R.; Zhao, W. X.; Zhang, F.; Zhang, D.; and Gai, K. 2024. Parrot: Enhancing Multi-Turn Instruction Following for Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9729–9750.

Team, Q. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Wang, G.; Cheng, S.; Zhan, X.; Li, X.; Song, S.; and Liu, Y. 2024a. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. In *The Twelfth International Conference on Learning Representations*.

Wang, Y.; Lipka, N.; Rossi, R. A.; Siu, A.; Zhang, R.; and Derr, T. 2024b. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19206–19214.

Wu, J.; Wang, C.; Su, T.; Haozhi, L.; JunYang, J.; Zhangchao, Z.; Pan, B.; SongpanYang, S.; Mingpeng, M.; Shi, K.; and Li, Z. 2025. Review-Instruct: A Review-Driven Multi-Turn Conversations Generation Method for Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Zhang, C.; Dai, X.; Wu, Y.; Yang, Q.; Wang, Y.; Tang, R.; and Liu, Y. 2025. A Survey on Multi-Turn Interaction Capabilities of Large Language Models. *arXiv preprint arXiv:2501.09959*.

Zhang, T.; Zhu, C.; Shen, Y.; Luo, W.; Zhang, Y.; Liang, H.; Yang, F.; Lin, M.; Qiao, Y.; Chen, W.; et al. 2024. Cfbench: A comprehensive constraints-following benchmark for llms. *arXiv preprint arXiv:2408.01122*.

Zhao, W.; Ren, X.; Hessel, J.; Cardie, C.; Choi, Y.; and Deng, Y. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 19724–19731.