Flow-guided Direct Preference Optimization for Knowledge Graph Reasoning with Trees

Tiesunlong Shen
Yunnan University, Kunming, China
Nanyang Technological University,
Singapore, Singapore
tensorshen@mail.ynu.edu.cn

Rui Mao*
Nanyang Technological University
Singapore, Singapore
rui.mao@ntu.edu.sg

Jin Wang
Yunnan University
Kunming, China
wangjin@ynu.edu.cn

Xuejie Zhang
Yunnan University
Kunming, China
xjzhang@ynu.edu.cn

Erik Cambria
Nanyang Technological University
Singapore, Singapore
cambria@ntu.edu.sg

Abstract

Recent advancements in knowledge graph question answering (KGQA) have shown promise, yet existing methods often fail to align with human reasoning patterns that involve continuous reflection and refinement. This paper proposes FD-PORT (flow-guided direct preference optimization for knowledge graph reasoning with trees), a novel approach that combines Monte Carlo Tree Search (MCTS) with flow-guided direct preference optimization (FDPO) for KGQA tasks. MCTS simulates human-like reasoning by systematically exploring multiple inference paths in knowledge graphs, while FDPO transforms the search feedback into fine-grained training signals through flow balance conditions. Unlike traditional methods focusing on end-to-end training or sequence-level preferences, FD-PORT establishes flow consistency between any states along the reasoning chain, enabling robust multi-hop reasoning that adapts to local decisions and long-range dependencies. Experimental results on three benchmark datasets demonstrate that FD-PORT significantly outperforms state-of-the-art methods, achieving up to 50.6% improvements over GPT-4 on complex multi-hop reasoning tasks with a smaller open-source language model. The framework is advanced in maintaining diverse reasoning paths while ensuring answer quality, closely mirroring human problem-solving strategies.

CCS Concepts

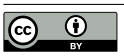
• **Computing methodologies** → **Natural language processing**; **Knowledge representation and reasoning**.

Keywords

Knowledge graph reasoning, Large language model, Direct preference optimization, Multi-hop question answering

ACM Reference Format:
Tiesunlong Shen, Rui Mao, Jin Wang, Xuejie Zhang, and Erik Cambria. 2025. Flow-guided Direct Preference Optimization for Knowledge Graph Reasoning with Trees. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729980

*Corresponding author.



SIGIR '25, July 13–18, 2025, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729980

1 Introduction

Reasoning about complex problems often relies on prior knowledge. Knowledge graphs, a widely used framework for knowledge representation, play a critical role in this domain [12]. One of the key advantages of knowledge graphs is their ability to support multistep reasoning, which is crucial for solving complex problems involving sequential dependencies [21]. Due to their tree-like structure, knowledge graphs allow for efficient traversal across multiple nodes and edges, enabling question-and-answering (QA) systems to derive insights through iterative reasoning over different paths. On the other hand, large language models (LLMs) can generate plausible responses that lack factual accuracy or logical coherence [28, 59], particularly when dealing with complex reasoning tasks that require a deeper understanding of dependencies and relationships between concepts (see Fig. 1). In light of this, a neurosymbolic approach that combines the generative and contextual understanding capabilities of LLMs, and the structured, interpretable reasoning power of knowledge graphs offers a promising direction to address the challenges of complex reasoning [5–7, 38].

Recently, three technical trends emerged by marrying knowledge graphs with LLMs, e.g., taking an LLM as a knowledge retriever [17, 53]; employing an LLM as a query generator [14, 42]; and directly tuning an LLM on knowledge graphs [27]. LLMs can effectively grasp common entities and relationships in knowledge graphs through extensive pre-training on large-scale world corpora. However, they are challenged by rare cases, indicating that LLMs do not effectively integrate their internal knowledge with external knowledge from knowledge graphs in a way that ensures reliable reasoning. These approaches do not follow the human intuitive process of continuously reflecting, adjusting, and integrating new information with existing knowledge. Humans iteratively reassess and modify their reasoning paths, dynamically incorporating new insights and correcting errors [29]. This continuous refinement and integration process is crucial for complex reasoning.

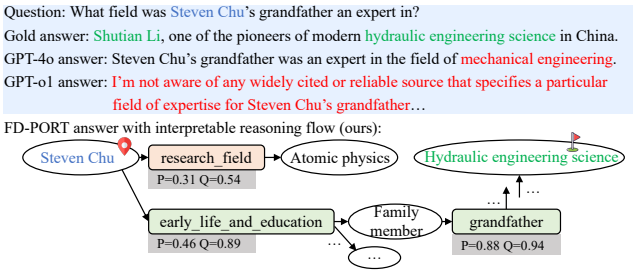


Figure 1: Different knowledge graph reasoning approaches.

We are motivated by this intuition, and integrate Monte Carlo Tree Search (MCTS) [11] with LLMs for knowledge graph reasoning. The proposed method integrates two specialized components, a policy model that guides path selection and an evaluation model that assesses state values. The MCTS component simulates human-like reasoning by systematically exploring multiple inference paths within the knowledge graph structure. As illustrated in Fig. 1, when answering a question about Steven Chu's grandfather's field of expertise, MCTS selects an appropriate relation to explore, based on two factors: the probability P predicted by the policy model indicating the likelihood of this relation leading to the correct answer; the Q-value derived from MCTS simulation reflecting the empirical quality of the path. This dual-guided exploration effectively identifies promising paths while avoiding misleading ones. Through this process, MCTS generates plausible solutions and creates a wealth of intermediate steps and decision points that reflect the subtle differences in the reasoning process, with these decision points forming a rich set of step-level preferences for model optimization.

However, effectively learning from these preferences is also challenging. Previous supervised MCTS-LLM methods rely on end-to-end training with large amounts of annotated data, using only the question and its final answer as a training pair [8, 44, 65]. This approach fails to capture the nuanced decision-making process required at each step of the reasoning chain, as it ignores the valuable information contained in intermediate reasoning steps. Recent advances in preference learning, particularly direct preference optimization (DPO) [37], have shown promise by introducing pairwise preference comparisons between solution sequences. While this allows for more fine-grained training signals than simple correct/incorrect labels, DPO still primarily operates at the sequence level - comparing complete reasoning paths rather than evaluating the quality of individual decisions at each step. This sequence-level optimization is insufficient for knowledge graph-based question-and-answering (KGQA) tasks, where humans typically maintain and evaluate multiple possible inference paths simultaneously, weighing their relative merits at each intermediate step. Current reward-maximizing approaches [34, 36], including DPO, often collapse to a limited set of fixed patterns, failing to capture this crucial aspect of human problem-solving. This mode collapse becomes particularly severe when the target distribution is misspecified, leading to overoptimized but unreliable solutions.

This work proposes **f**low-guided **d**irect **p**reference **o**ptimization for knowledge graph **r**easoning with **t**rees (FD-PORT) to bridge this gap. FD-PORT establishes an iterative learning loop between MCTS exploration and flow-guided direct preference optimization (FDPO). As shown in Fig. 2, for each state in the reasoning process, MCTS produces a quality estimate, while the evaluation model provides a value assessment. FDPO interprets the product of the MCTS estimate and the assessment as a unified reward signal to reflect both search feedback and learned value. Drawing inspiration from Generative Flow Networks (GFlowNets) [2], FDPO transforms these rewards into a flow that must be conserved between reasoning steps, ensuring that transitions leading to higher-reward steps receive proportionally more flow and thus prioritizing promising paths in the knowledge graph. Beyond adjacent steps, FDPO further enforces flow conservation across all subtrajectories along the reasoning chain, which expands preference signals from $O(1)$ in DPO or $O(n)$ in step-level-DPO to $O(n^2)$. While traditional DPO methods rely solely on final trajectory outcomes, FDPO captures comprehensive intermediate signals to maintain robust multi-hop reasoning that benefits from both search quality and evaluation alignment. Experiments on WebQuestionSP [63], Complex WebQuestions [52], and MetaQA [67] demonstrate the effectiveness of FD-PORT, as it achieves state-of-the-art performance across various reasoning scenarios with particularly strong improvements (22.4% compared to GPT-4) on complex multi-hop questions requiring 3-5 reasoning steps with an 8B parameter pre-trained language model (PLM). The results show that FD-PORT successfully combines the systematic exploration of MCTS with the flexibility of preference learning, enabling robust and interpretable knowledge graph reasoning.

The contributions of this work can be summarized as follows: (1) A novel KGQA framework that integrates MCTS with flow-guided preference optimization is proposed, enabling systematic exploration of reasoning paths while maintaining fine-grained trajectory alignment via an iterative learning paradigm that mirrors human reasoning processes. (2) A flow-guided direct preference optimization mechanism expands preference signals from $O(1)$ to $O(n^2)$ at sub-trajectory level, leveraging flow balance conditions to establish comprehensive intermediate state alignment. By enforcing flow conservation between any states along the reasoning chain, this mechanism enables the model to learn from all possible sub-trajectories rather than just terminal states, preventing the typical failure mode of overfitting to final answers while ignoring intermediate reasoning steps. (3) Extensive experiments suggest that FD-PORT achieves state-of-the-art performance on KGQA tasks, showing particular strength in complex multi-hop scenarios with improvements of up to 50.6% compared to GPT-4. The experiments validate the effectiveness of the integrated approach in maintaining consistent value propagation across different depths of reasoning while enabling systematic exploration of knowledge graphs.

2 Related Work

Knowledge Graph Question Answering. Multi-hop KGQA requires traversing multiple edges in knowledge graphs to locate answer entities that may be several hops away from subject entities [12]. Traditional approaches include embedding-based methods like KV-Mem [31] and EmbedKGQA [41] that leverage graph structure for answer prediction, retrieval-augmented methods such as GraftNet [48] and PullNet [47] that extract relevant subgraphs, and semantic parsing methods like SPARQL [50] and QGG [19] that generate structured queries. However, these methods rely heavily on large amounts of annotated data to learn fixed reasoning patterns,

and often struggle with maintaining reasoning consistency across multiple hops and lack the ability to dynamically adjust their inference paths based on intermediate results. FD-PORT addresses this limitation through MCTS-guided exploration that enables flexible reasoning path generation with reduced supervision requirements.

Monte Carlo Planning in Language Tasks. While Monte Carlo planning has achieved great success in strategic games [46], its application in natural language processing remains limited. Previous works explored its use in text generation [18] and decoding optimization [60]. FD-PORT extends these efforts by reformulating KGQA as a decision-making problem where MCTS guides the exploration of reasoning paths, similar to human problem-solving strategies [29]. However, these MCTS methods [43, 44] struggle to identify promising intermediate states during exploration, particularly in knowledge graph contexts where valuable trajectory information remains unexploited. FD-PORT addresses this challenge via flow-guided preference that systematically captures MCTS-generated signals, enabling more efficient path exploration.

LLM-based Reasoning. Recent advances demonstrate LLMs' reasoning capabilities through intermediate step generation [3, 22, 24, 58, 60]. However, issues such as hallucinations and factual inconsistency persist [13, 56]. ReACT [61] and Entailer [51] tackle these challenges by adding external knowledge and verification steps. However, local consistency often fails to ensure globally accurate outcomes, as their reasoning chains remain plausible but ungrounded due to the absence of structured guidance. FD-PORT addresses this challenge by integrating MCTS-guided knowledge graph traversal with preference optimization, ensuring each generated inference step is verified against structured knowledge while maintaining reasoning coherence.

3 Preliminaries: Flow Balance in Reasoning

GFlowNets establish flow balance as a fundamental principle: each state must maintain equilibrium between incoming and outgoing flows, while terminal state flows should be proportional to their rewards [2]. For any intermediate state s:

$$\sum_{s' \in \mathrm{Pred}(s)} F(s' \to s) = \sum_{s'' \in \mathrm{Succ}(s)} F(s \to s''), \tag{1}$$

where $F(s \to s')$ is the flow from state s to s'. At terminal states, the flow should be proportional to the reward: $F(s_{\text{terminal}}) \propto R(s_{\text{terminal}})$.

Traditional KGQA approaches primarily focus on accuracy in the final answer, which overlooks useful intermediate reasoning steps. FD-PORT extends this balancing concept to multi-hop scenarios by enforcing flow consistency between any two states along the reasoning chain. For states s_m and s_n in a reasoning path, their transition probability follows the flow ratio:

$$P(s_m \to s_n) \;\propto\; \frac{F(s_n)}{F(s_m)}. \tag{2}$$

This flow-guided mechanism naturally encourages transitions toward higher-reward states while maintaining consistency across multiple hops.

4 Methodology

FD-PORT integrates MCTS with FDPO through an iterative learning loop. MCTS explores knowledge graphs by treating entities as states and relations as actions, generating quality estimates (Q-values) for each visited state through systematic search. These Q-values naturally serve as reward signals for subsequent optimization - states with higher Q-values receive larger rewards, creating a principled way to evaluate the quality of different reasoning paths.

The flow-guided optimization framework then transforms these rewards into concrete training objectives through flow balance conditions. As illustrated in Fig. 2, this framework employs dual models - a policy model that guides path exploration and an evaluation model that assesses state values. By establishing flows proportional to the rewards between states, the framework ensures transition probabilities follow principled ratios determined by both search feedback and learned value estimates. The evaluation model maintains consistency between forward and backward flows across multiple hops, while the policy model learns to generate reasoning paths satisfying these flow balance conditions.

Through this integration, FD-PORT extends standard KGQA with principled flow constraints. The state s_t at each hop t represents the current reasoning chain prefix, with transitions governed by both MCTS-derived rewards and global flow balance requirements. This design enables continuous refinement - better search results lead to more accurate reward signals, which in turn guide the policy toward more promising reasoning paths, creating a virtuous cycle of improvement.

4.1 Flow-guided Monte Carlo Tree Search

Given a question q in a KGQA task, following [41], topic entity in q is mapped to a node v_q in knowledge graph G via entity-linking [62]. FD-PORT then transforms the problem into a Markov Decision Process and employs MCTS for solutions. MCTS explores the knowledge graph, representing states s_t as nodes and actions a_t as entity relations. Starting from the question entity s_0, as shown in Fig. 2, MCTS begins from the initial state and explores the knowledge graph by selecting actions (relations) with the help of pre-trained LLM to transition between states. The transition probability is set to 1 since the transition to the related entity is deterministic once the relation is selected. The objective is to discover a path leading to the correct answer entity for q. The state s_t at hop t is defined as the prefix of the reasoning chain. A new action a as well as the relation between two entities transitions the state s_t to s_{t+1}. MCTS begins iterates through the following four stages.

Selection. The selection phase identifies promising nodes for expansion using a hybrid strategy that combines adaptive exploration with probabilistic action selection. The strategy introduces an exploration probability $p_s \in [0, 1]$ and samples actions by:

$$\begin{aligned}\pi(a \mid s_t) =& (1 - p_s)\,\mathrm{Softmax}\left(Q(s_t, \cdot) + c_{puct}\, P(\cdot \mid s_t)\sqrt{\frac{N(s_t)}{1 + N(s_t, \cdot)}}\right) \\ &+ p_s\, \mathcal{U}(C(s_t)),\end{aligned} \tag{3}$$

where $Q(s_t, \cdot)$ represents the estimated state-action values, and $P(\cdot \mid s_t) = \pi_\theta(\cdot|x, s_t)$ is the prior probability from a PLM serving as the policy model.

The parameter c_{puct} controls the exploration-exploitation trade-off, while $N(s_t)$ and $N(s_t, \cdot)$ denote the visit counts of the state and state-action pairs, respectively. $\mathcal{U}(C(s_t))$ defines a uniform

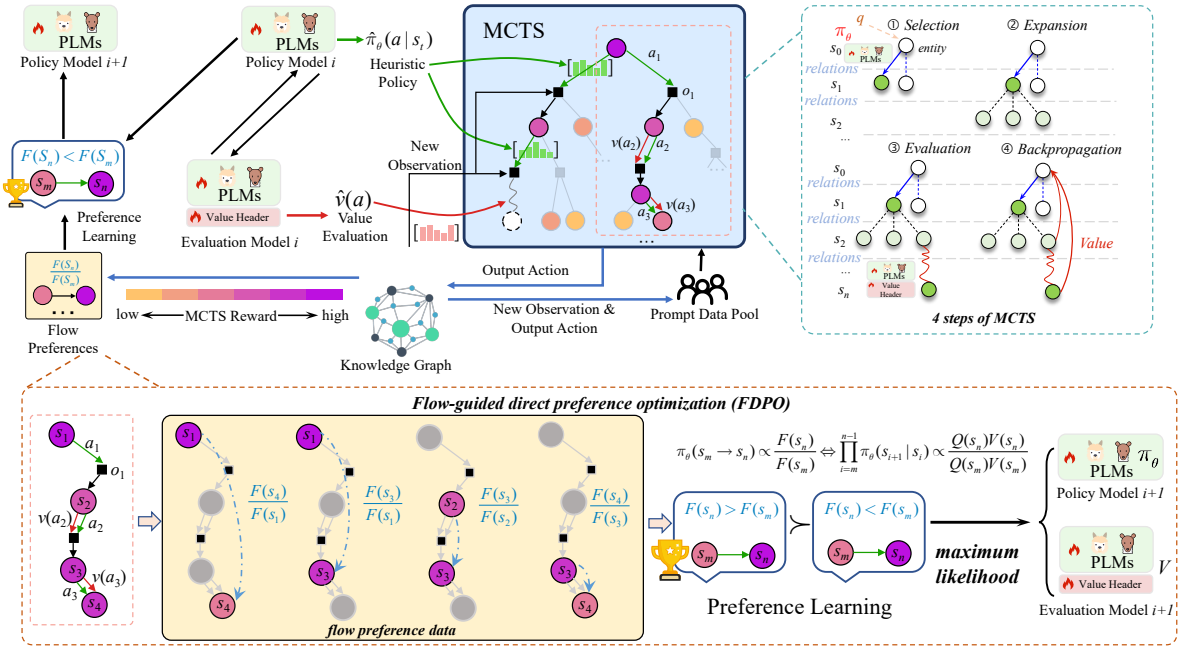


Figure 2: The overall framework of FD-PORT. The color depth represents the estimated value Q from MCTS simulation, with darker colors indicating higher Q and more preferred next hops. These estimated values serve as labels for sub-trajectory flow preferences to fine-tune both the policy and evaluation model.

distribution over the set of available actions $C(s_t)$ at state s_t. The exploration probability p_s is given by:

$$p_s = \frac{\varepsilon}{\log\left(N(s_t)+1\right)+1} \tag{4}$$

with ε is an exploration hyperparameter. This strategy combines softmax sampling with PUCT exploration. The policy π is a mixture of two components: with probability $(1-p_s)$, it samples actions from a softmax distribution over the combined score of Q-values and PUCT terms, where the PUCT term encourages the exploration of promising but less-visited actions; with probability p_s, it selects actions uniformly at random to ensure the thorough exploration of the action space. This adaptive mechanism gradually shifts the focus from exploration to exploitation as visit counts increase, while maintaining the capability for random exploration via the uniform distribution term. Then, the action is sampled by $a_t \sim \pi(\cdot \mid s_t)$.

Expansion and Evaluation. The expansion phase integrates newly discovered nodes into the search tree by generating all possible actions from the selected leaf node. For computational efficiency, evaluation employs a one-step rollout strategy. If the rollout action is terminal, the final answer is evaluated for correctness. The evaluation process is tailored to different phases: during training, it relies on ground truth rewards derived from the answer correctness; during inference, the evaluation model estimates state values.

The value function during training is defined as:

$$V(s) = I_{terminal}(a) \cdot r(s,a), \tag{5}$$

where $I_{terminal}$ indicates terminal actions and $r(s,a)$ is the final reward (1 for correct answers, 0 otherwise).

Backpropagation. The evaluation results are propagated back through the tree, updating statistics for each state-action pair (s', a') along the path:

$$N(s',a') \leftarrow N(s',a') + 1, \tag{6}$$

$$Q(s',a') \leftarrow Q(s',a') + \frac{1}{N(s',a')}(V(s) - Q(s',a')). \tag{7}$$

The Q-values obtained through MCTS simulations combine with value estimates V_ϕ from the evaluation model to define the total reward $R(s) = Q(s)V_\phi$ for each visited state. Following the GFlowNet principle that flow must match rewards at terminal states, these rewards directly determine the flow F(s) = R(s) through each state. As visualized in Fig. 2, states with higher combined rewards (shown in darker colors) receive larger flows, naturally establishing preferences over different reasoning paths.

4.2 Flow-guided Detailed Balance Optimization

Building upon the state-level flow preferences, FD-PORT implements a comprehensive flow-guided optimization mechanism that extends beyond adjacent state pairs. For a reasoning chain of states $\{s_1, s_2, \ldots, s_n\}$ discovered by MCTS, the flow balance constraints encompass any sub-trajectory $(s_m, s_{m+1}, \ldots, s_n)$, where $m < n$. This formulation through GFlowNets ensures that transition probabilities between states remain proportional to their flow ratios, creating a globally consistent preference optimization framework that considers both local transitions and long-range dependencies.

4.2.1 Model Architecture. FD-PORT employs a pre-trained LLM as the shared backbone network with two specialized output heads: a

policy head π_θ and an evaluate head V_ϕ. This design enables efficient parameter sharing while maintaining distinct functionalities for action selection and state evaluation.

4.2.2 FDPO Objective. The overall FDPO loss is defined as follows:

$$\mathcal{L}_{\text{FDPO}} = \mathcal{L}_{\text{policy}} + \lambda \mathcal{L}_{\text{evaluate}}, \tag{8}$$

where λ is a hyperparameter balancing the flow-guided policy loss ($\mathcal{L}_{\text{policy}}$) and the evaluation loss ($\mathcal{L}_{\text{evaluate}}$).

Flow-guided Policy Loss. The core mechanism of flow-guided policy optimization is to ensure that the transition probability between any two states along a reasoning path is proportional to their flow ratio: $P(s_m \to s_n) \propto F(s_n)/F(s_m)$. Therefore, FD-PORT employs flow conservation - the total flow entering each state must equal the flow leaving it. For any state s in the reasoning process, this fundamental balance equation is given by:

$$F(s) = \sum_{s' \in \text{Pred}(s)} P(s' \to s)\, F(s'), \tag{9}$$

where $\text{Pred}(s)$ denotes the set of predecessor states that can transition to s, directly reveals that increasing a state's flow $F(s)$ requires proportionally larger transition probabilities $P(s' \to s)$ from its predecessors. This mechanism naturally creates a "flow gain" effect: when a path leads to a higher-flow state, all transitions along that path must amplify their probabilities to maintain flow balance, thereby directing the policy toward more valuable outcomes.

For a trajectory $\tau : s_0 \to s_1 \to \cdots \to s_n$ in the knowledge graph, this balance requirement is formalized through the subtrajectory balance (SubTB) principle, which considers all possible partial paths from index m to n ($0 \le m < n \le$ trajectory length). Given flow function $F(\cdot)$, forward policy $\hat{\pi}_\theta$, and backward policy $\hat{\pi}_B$, the balance condition requires that the forward and backward flows match for each subtrajectory:

$$F(s_n) \prod_{i=m}^{n-1} \hat{\pi}_\theta(s_{i+1} \mid s_i) \;=\; F(s_m) \prod_{i=m}^{n-1} \hat{\pi}_B(s_i \mid s_{i+1}). \tag{10}$$

Taking logarithms of both sides and minimizing the squared difference between them yields the subtrajectory-balance loss:

$$\mathcal{L}_{\text{DB}}^{(\text{SubTB})} = \sum_{\tau \in \mathcal{D}} \sum_{m=0}^{n-1} \big[\Delta F_{m,n} - \Delta \pi_{m,n}\big]^2, \tag{11}$$

where $\Delta F_{m,n}$ represents the log-ratio of flow values:

$$\Delta F_{m,n} = \log F(s_n) - \log F(s_m). \tag{12}$$

$\Delta \pi_{m,n}$ denotes the cumulative log-ratio of forward and backward policies along the subtrajectory:

$$\Delta \pi_{m,n} = \sum_{i=m}^{n-1} [\log \hat{\pi}_B(s_i \mid s_{i+1}) - \log \hat{\pi}_\theta(s_{i+1} \mid s_i)]. \tag{13}$$

FD-PORT defines the flow function as the product of MCTS quality estimates and evaluation model values $F(s) = Q(s)\, V_\phi(s)$. Substituting $F(s_k) = Q(s_k)\, V_\phi(s_k)$ into the subtrajectory condition yields:

$$Q(s_n)\, V_\phi(s_n) \prod_{i=m}^{n-1} \hat{\pi}_\theta(s_{i+1} \mid s_i) \;=\; Q(s_m)\, V_\phi(s_m) \prod_{i=m}^{n-1} \hat{\pi}_B(s_i \mid s_{i+1}). \tag{14}$$

MCTS-guided KGQA employs a forward policy for search and inference, with backward transitions only serving training purposes. Following the previous work [2], this allows the adoption of a uniform backward policy ($\hat{\pi}_B(\cdot) = 1$) without compromising reasoning performance, leading to a simplified balance condition:

$$Q(s_n)\, V_\phi(s_n) \prod_{i=m}^{n-1} \hat{\pi}_\theta(s_{i+1} \mid s_i) \;=\; Q(s_m)\, V_\phi(s_m). \tag{15}$$

Hence, taking logarithms and penalizing the deviation from zero in squared form yields the subtrajectory-balance loss:

$$\mathcal{L}_{DB}(\hat{\pi}) \;=\; \sum_{(s_m \to \ldots \to s_n)} \Big(\Delta R_{m,n} - \Delta \pi_{m,n}\Big)^2, \tag{16}$$

where $\Delta R_{m,n} = \log\big[Q(s_n)\, V_\phi(s_n)\big] - \log\big[Q(s_m)\, V_\phi(s_m)\big]$ represents the log-ratio of quality-value products between states s_n and s_m. $\Delta \pi_{m,n} = \sum_{t=m}^{n-1} \log \frac{\hat{\pi}(s_{t+1}|s_t)}{\hat{\pi}_B(s_t|s_{t+1})}$ denotes the cumulative log-ratio of forward and backward policies along the subtrajectory.

To obtain an equivalent expression under the flow-balance constraint, the symmetry in this equality enables a systematic rearrangement of terms. Dividing both sides by $Q(s_m)\, V_\phi(s_m)$ and $Q(s_n)\, V_\phi(s_n)$ isolates $Q(s_m)\, V_\phi(s_n)$ and $Q(s_n)\, V_\phi(s_m)$ in a ratio while preserving the underlying relationship. This leads to:

$$\log \frac{Q(s_m)\, V_\phi(s_n)}{Q(s_n)\, V_\phi(s_m)} \;=\; \sum_{i=m}^{n-1} \log \hat{\pi}_\theta(s_{i+1} \mid s_i), \tag{17}$$

which then penalizes again in squared form over all subtrajectories $\big(s_m \to \cdots \to s_n\big)$. Concretely, the final flow-guided policy loss is:

$$\mathcal{L}_{policy}(\hat{\pi}) = \sum_{\tau \in \mathcal{D}} \sum_{m<n} \left(\log \frac{Q(s_m)\, V_\phi(s_n)}{Q(s_n)\, V_\phi(s_m)} - \sum_{i=m}^{n-1} \log \hat{\pi}\big(s_{i+1} \mid s_i\big) \right)^2. \tag{18}$$

This rearranged formulation ensures each partial path respects the flow implied by MCTS quality estimates and evaluation model values. The forward policy $\hat{\pi}$ learns to favor sub-paths leading to states with higher combined quality-value assessments in multi-hop reasoning, enhancing the overall search quality.

Evaluation Loss. It ensures that the evaluation head accurately predicts the relative quality of different actions for a given hop-level preference instance $(s_t, a_w \succ a_l)$, where a_w and a_l represent the preferred and dispreferred actions (relations), respectively.

$$\mathcal{L}_{\text{evaluate}} = \max(0, \gamma - (V_\phi(s_t, a_w) - V_\phi(s_t, a_l))), \tag{19}$$

where γ is a margin hyperparameter encouraging a separation between the values of preferred and dispreferred actions.

4.2.3 Training Procedure. The training procedure alternates between three steps:

1) *Path Generation*: Use the current policy model $\hat{\pi}_\theta$ to perform MCTS and generate reasoning paths $\{(s_0^{(i)}, a_0^{(i)}, ..., s_T^{(i)})\}_{i=1}^N$ with MCTS estimate $Q(s_t)$. 2) *Flow Computation*: For each state in the generated paths, compute the evaluation model estimation $V(s_t)$, the overall reward $R(s_t)$ and flow ratios between consecutive states. 3) *Policy Update*: Update the policy parameters θ by minimizing $\mathcal{L}_{FDPO}$ using gradient descent $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{FDPO}$, where α is the learning rate.

The above process ensures that the policy model learns to generate diverse reasoning paths, while maintaining high answer quality and consistency between the policy and evaluate heads.

4.3 Inference

After training the policy and evaluating models, FD-PORT adopts a more efficient hop-level beam search (HBS) for inference to reduce computational costs while maintaining reasoning quality. Given beam sizes B_1 and B_2, HBS maintains a set of B_1 most promising states at each reasoning step, denoted as $C = \{s_t\}$. For each state $s_t \in C$, the policy head π_θ generates B_2 candidate actions in parallel, enabling efficient exploration of multiple reasoning paths, simultaneously. Each resulting state-action pair is then evaluated by the evaluate head V_ϕ, producing a combined score based on both policy probability and value estimate. This approach eliminates the need for full tree construction while preserving the guidance from both learned models. Among all $B_1 \times B_2$ candidates, the top B_1 states with the highest scores are retained to form the candidate set for the next hop. This process iterates until it either arrives at a terminal state or reaches the predefined maximum hop limit.

5 Experiment

5.1 Datasets

Following the previous work on KGQA [15], we evaluated the reasoning performance of FD-PORT on three benchmark KGQA datasets, namely WebQuestionSP (WebQSP) [63], Complex WebQuestions (CWQ) [52], and MetaQA [67]. WebQSP and CWQ are both based on the Freebase [4] knowledge graph, with CWQ representing particularly challenging scenarios requiring up to 4-hop reasoning chains and complex logical operations, while WebQSP contains relatively simpler 2-hop reasoning problems. MetaQA complements the evaluation with diverse domain coverage spanning movies, TV shows, and music, featuring reasoning chains of up to 3 hops. These datasets can evaluate multi-hop reasoning capabilities, with varying levels of complexity and different knowledge domains.

5.2 Baselines

The following baseline methods are compared in our experiments: **1) Traditional Embedding & Semantic Parsing methods** that leverage graph structure and semantic information for question answering: KV-Mem [31], EmbedKGQA [41], NSM [10], TransferNet [45], KGT5 [40], SPARQL [50], QGG [19]. **2) Retrieval methods** that focus on extracting relevant subgraphs or paths from knowledge graphs: GraftNet [48], PullNet [47], SR+NSM [66], SR+NSM+E2E [66]. **3) Direct reasoning with LLMs** that utilizes LLMs' inherent reasoning capabilities without explicit graph structure: Davinci-003 [35], Llama2-13B [55], Alpaca-7B [54], GPT3.5 [32], GPT4 [33], AgentBench [25], ChatGPT+CoT [58]. **4) LLMs + knowledge graph methods** that combine the strengths of both LLMs and structured knowledge graphs for enhanced reasoning: KB-Binder [20], KAPING [1], KD-CoT [57], UniKGQA [16], DECAF (DPR+FiD-3B) [64], StructGPT [14], ReasoningLM [15], RoG [27], ToG-ChatGPT [49], EtD-ChatGPT [23], Readi- [9], Readi-GPT4 [9].

5.3 Setups

Policy Model and Evaluation Model. For these two modules of FD-PORT, we use LLaMA3-Base-8B [30] as the base PLM. We iteratively train this model on the training splits of three datasets as described in Section 4.2.3. The training employs a learning rate of 4e-5 and the AdamW optimizer [26]. For the value head in evaluation model, we add a single linear layer with tanh activation on top of the LLM's hidden states. The evaluation value is predicted only for the last token of each reasoning step.

Training Data Generation via MCTS. We iteratively generate data, train our policy, and evaluate the models over $R = 4$ rounds, continuing this process until the improvement observed between consecutive rounds becomes marginal. In every round, we build 8 trees for each question-answer pair and randomly sample at most 5 correct and 5 incorrect solution processes. We then extract hop-level preferences from the trees in a top-down manner. The ratio between positive and negative examples is about $1:1$. The search breadth is set to $b_1 = 50$ and $b_2 = 5$.

5.4 Main Results

Table 1 shows the evaluation of FD-PORT against various baselines across three benchmark datasets, yielding an average Hits@1 accuracy of 92.4%. On WebQSP and CWQ datasets, FD-PORT achieves 89.2% and 74.5% Hits@1 accuracy, respectively, surpassing the previous best results from RoG (85.7%) and ReasoningLM (69.0%). This improvement stems from the FDPO mechanism, enabling more effective exploration of reasoning paths while maintaining consistency between local decisions and global objectives. The performance advantage is particularly evident on CWQ, highlighting the effectiveness of the sub-trajectory balance framework in handling complex multi-hop reasoning scenarios. For the MetaQA dataset with different hops (M-1, M-2, and M-3 in Table 1), FD-PORT demonstrates consistent performance across different reasoning depths, achieving 98.7%, 99.7%, and 99.8% accuracy on three MetaQA datasets respectively. These results validate the scalability of the flow-guided optimization approach in maintaining reasoning quality as path length increases.

Notably, FD-PORT achieves these results using only LLaMA3-Base-8B as the backbone model, yet surpasses trillion-parameter proprietary models in performance. While GPT-4 achieves 70.7% on WebQSP and 52.1% on CWQ, FD-PORT improves these results by 18.5% and 22.4%, respectively. This performance gap demonstrates that integrating structured knowledge graph exploration with flow-guided preference learning can effectively address LLMs' limitations in multi-hop reasoning without requiring massive model parameters. Additionally, FD-PORT performs competitively against other knowledge graph-enhanced LLM methods, showing consistent improvements over recent approaches such as EtD-ChatGPT and Readi-GPT4.

5.5 Effectiveness Analysis

The effectiveness of FD-PORT manifests through two complementary analyses that examine its decision-making capabilities across different reasoning stages. Two baselines are compared: SFT+MCTS employs standard supervised fine-tuning (SFT) with MCTS, focusing on direct imitation of correct reasoning paths, while DPO+MCTS

Table 1: Performance comparison with different baselines on the three KGQA datasets, measured by Hit@1 accuracy (%). The Avg. column shows the mean performance across all available datasets for each method. KGs denote knowledge graphs. M-x denotes the MetaQA dataset with different hops.

Methods	WebQSP	CWQ	M-1	M-2	M-3	Avg.
Embedding &Semantic Parsing-based Baselines						
KV-Mem [31]	46.7	18.4	96.2	82.7	48.9	58.6
EmbedKGQA [41]	66.6	-	97.5	98.8	94.8	89.4
NSM [10]	67.7	47.6	97.1	99.9	98.9	82.2
TransferNet [45]	71.4	48.6	97.5	**100**	**100**	83.5
KGT5 [40]	56.1	36.5	-	-	-	46.3
SPARQL [50]	-	31.6	-	-	-	31.6
QGG [19]	73.0	36.9	-	-	-	55.0
Retrieval-based Baselines						
GraftNet [48]	66.4	36.8	97.0	94.8	77.7	74.5
PullNet [47]	68.1	45.9	-	-	-	57.0
SR+NSM [66]	69.5	50.2	-	-	-	59.9
SR+NSM+E2E [66]	69.5	49.3	-	-	-	59.4
LLM-based Baselines						
Davinci-003 [35]	48.7	-	52.1	25.3	42.5	42.2
Llama2-13B [55]	40.9	22.1	31.9	15.8	34.9	29.1
Alpaca-7B [54]	51.8	27.4	-	-	-	39.6
GPT3.5 [32]	65.7	44.7	61.9	31.0	43.2	49.3
GPT4 [33]	70.7	52.1	71.8	52.5	49.2	59.3
AgentBench [25]	47.8	24.8	-	-	-	36.3
ChatGPT+CoT [58]	75.6	48.9	-	-	-	62.3
LLM- and KG-based Baselines						
KB-Binder [20]	74.4	-	92.9	99.9	99.5	91.7
KAPING [1]	73.9	-	-	-	-	73.9
KD-CoT [57]	68.6	55.7	-	-	-	62.2
UniKGQA [16]	75.1	50.7	97.5	99.0	99.1	84.3
DECAF [64]	82.1	-	-	-	-	82.1
StructGPT [14]	72.6	-	94.2	93.9	80.2	85.2
ReasoningLM [15]	78.5	69.0	96.5	98.3	92.7	87.0
RoG [27]	85.7	62.6	-	-	84.8	77.7
ToG-ChatGPT [49]	76.2	58.9	-	-	-	67.6
EtD-ChatGPT [23]	82.5	62.0	98.1	99.7	99.7	88.4
Readi-GPT3.5 [9]	74.3	55.6	98.4	99.9	99.4	85.5
Readi-GPT4 [9]	78.7	67.0	98.5	99.9	99.2	88.7
FD-PORT (Ours)	**89.2**	**74.5**	**98.7**	99.7	99.8	**92.4**

extends this approach by incorporating sequence-level preferences between complete reasoning chains. In contrast, FD-PORT introduces fine-grained subtrajectory balance constraints that provide detailed feedback at each intermediate state through FDPO.

Figure 3a presents the distribution of Q-value differences ($\Delta Q = Q_{correct} - Q_{wrong}$) between correct and incorrect next-hop selections, where $Q_{correct}$ represents the Q-values of states leading to correct final answers, and Q_{wrong} denotes those leading to incorrect answers. For each next-hop selection, a $Q_{correct}$ and a Q_{wrong} state are randomly sampled. This metric directly reflects a method's ability to discriminate between promising and suboptimal choices during the reasoning process - a larger positive ΔQ indicates a stronger preference for correct paths over incorrect ones, while values near zero suggest difficulty in distinguishing between different options. SFT+MCTS exhibits a relatively broad distribution centered near zero, indicating limited discrimination between correct and

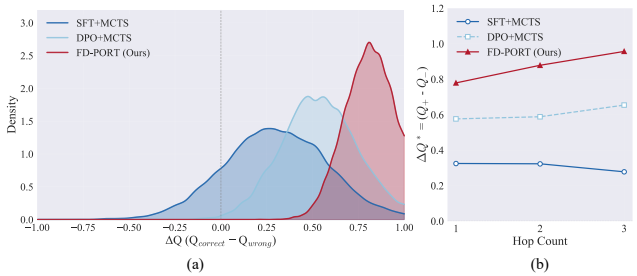


Figure 3: Performance comparison of MCTS-based methods on the CWQ validation set: (a) Distribution of ΔQ (the difference between Q-values of correct and incorrect paths) for three different approaches. (b) ΔQ^* values across increasing hop counts (1-3), where ΔQ^* quantifies the separation between positive and negative Q-values. Higher ΔQ^* indicates better ability to distinguish between correct and incorrect reasoning paths.

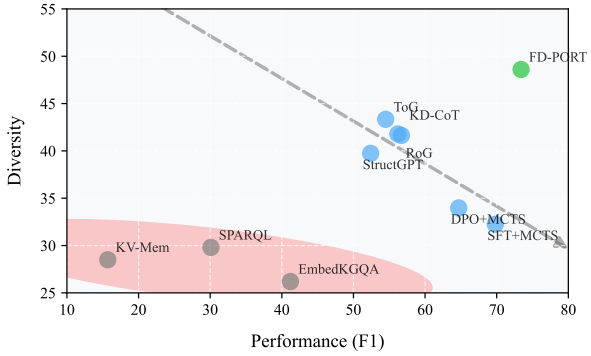


Figure 4: Performance-diversity trade-off analysis across different KGQA methods, based on the CWQ testing set. The lower-left region indicates methods with fundamental limitations in both aspects, while the upper-right region represents methods achieving both high performance and diversity.

incorrect choices due to its reliance solely on supervised signals. While DPO+MCTS shows improved separation with its distribution shifted toward positive values through sequence-level preference learning, FD-PORT achieves the most pronounced separation, with its distribution significantly skewed toward higher positive values and minimal overlap in the negative region. This enhanced discriminative power stems from the FDPO mechanism, which establishes consistent value assessments across different reasoning depths through fine-grained subtrajectory balance constraints.

Figure 3b further validates this advantage by tracking the evolution of Q-value gaps ($\Delta Q^* = Q_+ - Q_-$) across increasing hop counts. With Q-values bounded in (0,1) representing estimated success probabilities, Q_+ denotes the average Q-values typically above 0.5 for promising actions, while Q_- captures those generally below 0.5 for less promising ones. The three methods exhibit distinct patterns in their discriminative capabilities across multiple reasoning steps. SFT+MCTS, relying solely on supervised signals, starts with low ΔQ^* values and shows only marginal improvement at the second hop through MCTS updates before deteriorating at the third hop due to accumulated error paths. DPO+MCTS demonstrates higher overall ΔQ^* values through sequence-level preference optimization

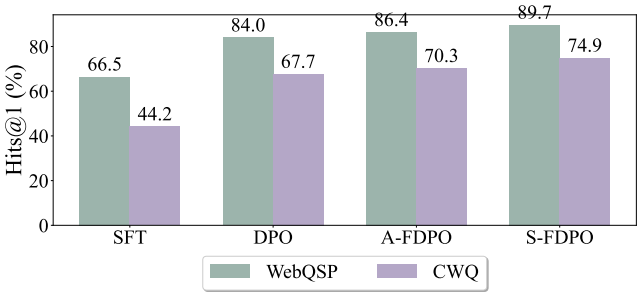


Figure 5: Different fine-tuning method performance on the WebQSP and CWQ validation sets: Supervised fine-tuning (SFT), direct preference optimization (DPO), adjacent flow DPO (A-FDPO), and subtrajectory flow DPO (S-FDPO).

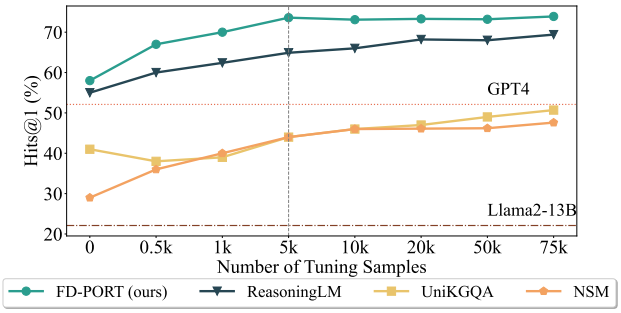


Figure 6: The Hits@1 scores of FD-PORT against three strong baselines (i.e., ReasoningLM, UniKGQA, and NSM) on CWQ validation sets when tuning with various numbers of samples

with MCTS updates, but shows only modest gains before the final hop due to the absence of fine-grained intermediate supervision. In contrast, FD-PORT achieves higher ΔQ^* values from the first hop (0.39) and maintains steady growth through the third hop (0.47), benefiting from the synergy between subtrajectory flow balance and preference optimization. This continuous improvement demonstrates how flow-guided optimization accumulates and reinforces correct decision preferences while increasingly penalizing suboptimal choices at each step, enabling FD-PORT to maintain and expand its discriminative advantage even in complex multi-hop scenarios.

5.6 Balance between Accuracy and Diversity

This section examines the trade-off between reasoning accuracy and path diversity across different KGQA methods. To quantify diversity, each method generates three distinct reasoning paths for every input query with diversity measured as the average cosine distance between SentenceBERT [39] embeddings of these generated paths. This metric captures semantic differences while being robust to variations in path length. The analysis plots these diversity scores against F1 performance on the CWQ dataset, as shown in Figure 4.

Most LLM-enhanced methods, including baseline variants SFT+MCTS and DPO+MCTS, exhibit a clear trade-off between accuracy and diversity - as F1 scores increase, diversity tends to decrease. While SR+NSM and StructGPT achieve reasonable F1 scores, their diversity metrics remain constrained due to their tendency to converge on a limited set of reasoning patterns. SFT+MCTS shows similar limitations due to its direct imitation learning approach. Although DPO+MCTS achieves slightly improved diversity through sequence-level preference optimization, it still cannot overcome this fundamental trade-off. In contrast, FD-PORT breaks this trade-off by employing flow-guided MCTS to systematically explore multiple reasoning paths rather than greedily pursuing a single high-reward trajectory. The detailed balance optimization only requires positive flow gains between states as a flexible preference direction, rather than enforcing strict end-to-end correctness constraints, enabling FD-PORT to reach significantly higher diversity while maintaining competitive F1 performance. In real-world applications, achieving accurate predictions with diverse path explorations on knowledge graphs is critical for complex reasoning, because diverse reasoning paths enable more comprehensive coverage of potential inference trajectories and reduce the risk of converging to suboptimal solutions in multi-hop reasoning scenarios [2]. Traditional Embedding & Semantic Parsing methods such as KV-Mem and EmbedKGQA cluster in the lower-left region, showing both limited diversity and modest F1 scores, primarily due to their reliance on memory network retrieval over documents or limited segments without explicit multi-path reasoning.

Table 2: Ablation studies on the CWQ validation set of various components of FD-PORT, including LLM as policy model (L-PM), MCTS, and evaluation model (EM).

L-PM	MCTS	EM	F1	Hits@1
✓	×	×	39.2	42.3
×	✓	✓	44.6	49.7
✓	✓	×	59.9	64.1
✓	✓	✓	68.2	74.9

5.7 Ablation Study

Our ablation studies examine two critical aspects of FD-PORT: the contribution of individual architectural components and the impact of different fine-tuning strategies. These experiments provide insights into how each element contributes to the overall reasoning capability and validate the effectiveness of FDPO.

Table 2 demonstrates the progressive performance gains on the CWQ validation set, achieved by incorporating different components. Using an LLM solely as a policy model yields modest results (F1: 39.2%, Hits@1: 42.3%), as the model relies entirely on pre-trained knowledge without structured exploration. While combining MCTS with the evaluation model improves the performance (F1: 44.6%, Hits@1: 49.7%), the absence of language model guidance limits the effectiveness of tree search in complex reasoning scenarios. Integrating the policy model with MCTS creates a more powerful synergy (F1: 59.9%, Hits@1: 64.1%), where language understanding guides the exploration of knowledge graphs. The complete FD-PORT framework achieves optimal performance (F1: 68.2%, Hits@1: 74.9%) by leveraging all components in concert, demonstrating how the evaluation model's value estimates complement policy-guided tree search for robust multi-hop reasoning.

Figure 5 shows the advantages of flow-guided optimization through a comparative analysis of fine-tuning approaches. Traditional SFT establishes initial performance levels on WebQSP and CWQ (66.5% and 44.2% Hits@1), indicating the limitations of direct

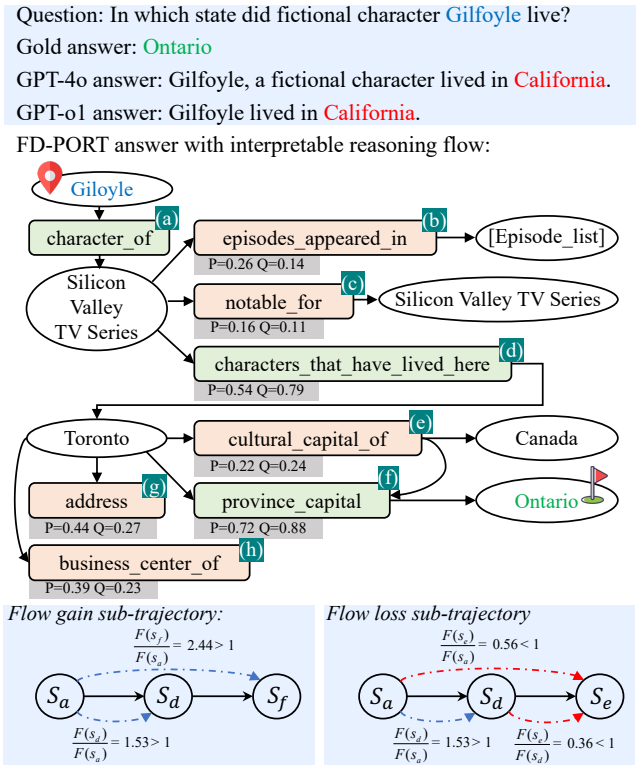


Figure 7: Case study of different reasoning methods.

imitation learning in complex reasoning tasks. The introduction of sequence-level preference optimization through DPO substantially improves performance (84.0% and 67.7% Hits@1) by capturing relative quality between complete reasoning paths. Adjacent flow direct preference optimization (A-FDPO), which enforces flow consistency between adjacent state transitions in reasoning chains, leads to more coherent step-by-step reasoning (86.4% and 70.3% Hits@1). Subtrajectory flow direct Preference optimization (S-FDPO), which maintains flow balance across all possible subtrajectories within the reasoning process, achieves peak performance (89.7% and 74.9% Hits@1) by effectively addressing both local transitions and global reasoning coherence. This systematic improvement across dataset validates the fundamental premise of flow-guided optimization: maintaining consistency across multiple reasoning scales is crucial for reliable knowledge graph question answering.

5.8 Efficiency Analysis

This section investigates the fine-tuning efficiency of FD-PORT compared to existing methods. As shown in Figure 6, FD-PORT achieves superior performance with significantly fewer training samples, reaching stable performance at just 5k samples. This efficiency gain stems from the FDPO mechanism that expands the effective training signals. Specifically, for a reasoning path containing n states, foundation models like GPT4 and Llama2-13B only utilize the final answer as a preference signal, yielding $O(1)$ training signal per reasoning chain. Methods like ReasoningLM improve upon this by incorporating intermediate state preferences, achieving $O(n)$ training signals. In contrast, FD-PORT enforces flow balance conditions between any two states along the reasoning chain, yielding $O(n^2)$ training signals through sub-trajectory level comparisons. This quadratic expansion of training signals enables more efficient learning from limited data. The empirical results also validate this theoretical advantage. While ReasoningLM requires the full 75k samples to reach its peak performance, FD-PORT achieves comparable results with just 5k samples and maintains stable performance.

5.9 Case study

To demonstrate the effectiveness of FD-PORT's reasoning process, we present a case study from the CWQ testing set, as illustrated in Figure 7. The question "In which state did fictional character Gilfoyle live?" exemplifies how FD-PORT combines MCTS exploration with flow-guided optimization to achieve accurate reasoning.

For clarity in presenting our method, we illustrate a subset of key paths explored by MCTS. FD-PORT processes the reasoning task by starting from the node of Gilfoyle, because it was explicitly mentioned in the question. Through relation (a), FD-PORT first identifies Gilfoyle as a character in Silicon Valley TV Series. Among relations (b), (c), and (d), FD-PORT selects relation (d) with the highest policy probability and Q-value (P=0.54, Q=0.79), leading to Toronto. From Toronto, among relations (e), (g), (h), and (f), FD-PORT follows relation (f) (P=0.72, Q=0.88) to reach Ontario. The flow ratios in the right panel of Fig. 7 validate this reasoning path, showing increasing confidence with flow gains with $F(s_d)/F(s_a) = 1.53$ and $F(s_f)/F(s_d) = 2.44$, indicating that FDPO training has successfully guided the model to assign higher confidence to the promising path (a)→(d)→(f). Conversely, the lower trajectory's deteriorating flow ratios ($F(s_e)/F(s_d) = 0.36$) reflect the training process has learned to downweight less relevant paths (a)→(d)→(e).

Through this FDPO-enhanced training, FD-PORT successfully identifies Ontario as the answer by effectively reasoning over the knowledge graph structure, while both GPT-4o and GPT-o1 incorrectly suggest California based solely on their pre-trained knowledge. The case illustrates how the flow-guided training strategy enables MCTS to develop more accurate state-value estimates and policy preferences specifically for knowledge graph traversal, leading to more reliable reasoning capabilities compared to pure language models that lack explicit graph-structured reasoning mechanisms.

6 Conclusion

This paper introduces FD-PORT, a novel framework that enhances KGQA through FDPO. By integrating MCTS exploration with subtrajectory flow balance constraints, FD-PORT enables more reliable multi-hop reasoning that aligns with both local decisions and global objectives. The experiments suggest that FD-PORT significantly outperforms state-of-the-art methods, particularly in complex reasoning scenarios, while requiring substantially less training data.

Acknowledgments

Tiesunlong Shen, Jin Wang and Xuejie Zhang are supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61966038 and 62266051. Rui Mao and Erik Cambria are supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

[1] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*. Association for Computational Linguistics, 78–106.

[2] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. 2023. GFlowNet foundations. *The Journal of Machine Learning Research* 24, 1 (2023), 10006–10060.

[3] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17682–17690.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[5] Erik Cambria, Rui Mao, Melvin Chen, Zhaoxia Wang, and Seng-Beng Ho. 2023. Seven pillars for the future of artificial intelligence. *IEEE Intelligent Systems* 38, 6 (2023), 62–69. https://doi.org/10.1109/MIS.2023.3329745

[6] Erik Cambria, Rui Mao, Sooji Han, and Qian Liu. 2022. Sentic Parser: A Graph-Based Approach to Concept Extraction for Sentiment Analysis. In *Proceedings of ICDM Workshops*. 413–420.

[7] Sandro Cavallari, Erik Cambria, Hongyun Cai, Kevin Chang, and Vincent Zheng. 2019. Embedding both finite and infinite communities on graphs. *IEEE Computational Intelligence Magazine* 14, 3 (2019), 39–50.

[8] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. AlphaMath Almost Zero: process Supervision without process. *arXiv preprint arXiv:2405.03553* (2024).

[9] Sitao Cheng, Ziyuan Zhuang, Yong Xu, Fangkai Yang, Chaoyun Zhang, Xiaoting Qin, Xiang Huang, Ling Chen, Qingwei Lin, Dongmei Zhang, et al. 2024. Call me when necessary: LLMs can efficiently and faithfully reason over structured environments. *arXiv preprint arXiv:2403.08593* (2024).

[10] Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 553–561.

[11] Steven James, George Konidaris, and Benjamin Rosman. 2017. An analysis of Monte Carlo Tree Search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[12] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems* 33, 2 (2021), 494–514.

[13] Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 1827–1843.

[14] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A general framework for large language model to reason over structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 9237–9251.

[15] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2023. ReasoningLM: Enabling structural subgraph reasoning in pre-trained language models for question answering over knowledge graph. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 3721–3735.

[16] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2023. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. arXiv:2212.00959 [cs.CL] https://arxiv.org/abs/2212.00959

[17] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023. KG-GPT: A general framework for reasoning on knowledge graphs using large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 9410–9421.

[18] Kaori Kumagai, Ichiro Kobayashi, Daichi Mochihashi, Hideki Asoh, Tomoaki Nakamura, and Takayuki Nagai. 2016. Human-like natural language generation using Monte Carlo Tree Search. In *Proceedings of the INLG 2016 Workshop on Computational Creativity in Natural Language Generation*. 11–18.

[19] Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

[20] Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6966–6980.

[21] Qika Lin, Jun Liu, Rui Mao, Fangzhi Xu, and Erik Cambria. 2023. TECHS: Temporal Logical Graph Networks for Explainable Extrapolation Reasoning. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1. Association for Computational Linguistics, Toronto, Canada, 1281–1293.

[22] Qika Lin, Rui Mao, Jun Liu, Fangzhi Xu, and Erik Cambria. 2023. Fusing topology contexts and logical rules in language models for knowledge graph completion. *Information Fusion* 90 (2023), 253–264.

[23] Guangyi Liu, Yongqi Zhang, Yong Li, and Quanming Yao. 2024. Explore then determine: A GNN-LLM synergy framework for reasoning over knowledge graph. *arXiv preprint arXiv:2406.01145* (2024).

[24] Qian Liu, Xiubo Geng, Yu Wang, Erik Cambria, and Daxin Jiang. 2024. Disentangled Retrieval and Reasoning for Implicit Question Answering. *IEEE Transactions on Neural Networks and Learning Systems* 35, 6 (2024), 7804–7815.

[25] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. [n. d.]. AgentBench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*.

[26] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[27] Linhao Luo, Yuan-Fang Li, Reza Haf, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

[28] Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2024. GPTEval: A survey on assessments of ChatGPT and GPT-4. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 7844–7866.

[29] Hugo Mercier and Dan Sperber. 2011. Why do humans reason? Arguments for an argumentative theory. *Behavioral and Brain Sciences* 34, 2 (2011), 57–74.

[30] MetaAI. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/ Accessed: 2024-06-15.

[31] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1400–1409.

[32] OpenAI. 2023. ChatGPT: Optimizing language models for dialogue. https://openai.com/blog/chatgpt Accessed: 2023-06-15.

[33] OpenAI. 2023. GPT-4: The next generation of AI language models. https://openai.com/research/gpt-4 Accessed: 2023-06-15.

[34] Yassine Ouali, Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. 2025. CLIP-DPO: Vision-language models as a source of preference for fixing hallucinations in LVLMs. In *European Conference on Computer Vision*. Springer, 395–413.

[35] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[36] Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2025. Safe-CLIP: Removing nsfw concepts from vision-and-language models. In *European Conference on Computer Vision*. Springer, 340–356.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[38] Dheeraj Rajagopal, Erik Cambria, Daniel Olsher, and Kenneth Kwok. 2013. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *WWW*. 565–570.

[39] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[40] Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2814–2828.

[41] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4498–4507.

[42] Tiesunlong Shen, Erik Cambria, Jin Wang, Yi Cai, and Xuejie Zhang. 2025. Insight at the right spot: Provide decisive subgraph information to Graph LLM with reinforcement learning. *Information Fusion* 117 (2025), 102860.

[43] Tiesunlong Shen, Jin Wang, Xuejie Zhang, and Erik Cambria. 2025. Hop-level Direct Preference Optimization for Knowledge Graph Reasoning with Trees. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[44] Tiesunlong Shen, Jin Wang, Xuejie Zhang, and Erik Cambria. 2025. Reasoning with Trees: Faithful Question Answering over Knowledge Graph. In *Proceedings of the 31st International Conference on Computational Linguistics*, Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 3138–3157. https://aclanthology.org/2025.coling-main.211/

[45] Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 4149–4158.

[46] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[47] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2380–2390.

[48] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4231–4242.

[49] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. [n. d.]. Think-on-Graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

[50] Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. 2020. SPARQA: skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8952–8959.

[51] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2078–2093.

[52] Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 641–651.

[53] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Can ChatGPT replace traditional KBQA models? An in-depth analysis of the question answering performance of the GPT LLM family. In *International Semantic Web Conference*. Springer, 348–367.

[54] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model.

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

[56] Boshi Wang, Xiang Yue, and Huan Sun. 2023. Can ChatGPT defend its belief in truth? Evaluating LLM reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 11865–11881.

[57] Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven CoT: Exploring faithful reasoning in LLMs for knowledge-intensive question answering. arXiv:2308.13259 [cs.CL] https://arxiv.org/abs/2308.13259

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances In Neural Information Processing Systems* 35 (2022), 24824–24837.

[59] Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are Large Language Models Really Good Logical Reasoners? A Comprehensive Evaluation and Beyond. *IEEE Transactions on Knowledge and Data Engineering* 37, 4 (2025), 1620–1634.

[60] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[61] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[62] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*.

[63] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 201–206.

[64] Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. 2023. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. arXiv:2210.00063 [cs.CL] https://arxiv.org/abs/2210.00063

[65] Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. 2024. LLaMA-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884* (2024).

[66] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5773–5784.

[67] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.