Towards Completeness-Oriented Tool Retrieval for Large Language Models

Changle Qu
Sunhao Dai
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{changlequ,sunhaodai}@ruc.edu.cn

Xiaochi Wei
Baidu Inc.
Beijing, China
weixiaochi@baidu.com

Hengyi Cai
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
caihengyi@ict.ac.cn

Shuaiqiang Wang
Baidu Inc.
Beijing, China
shqiang.wang@gmail.com

Dawei Yin
Baidu Inc.
Beijing, China
yindawei@acm.org

Jun Xu*
Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
{junxu,jrwen}@ruc.edu.cn

ABSTRACT

Recently, integrating external tools with Large Language Models (LLMs) has gained significant attention as an effective strategy to mitigate the limitations inherent in their pre-training data. However, real-world systems often incorporate a wide array of tools, making it impractical to input all tools into LLMs due to length limitations and latency constraints. Therefore, to fully exploit the potential of tool-augmented LLMs, it is crucial to develop an effective tool retrieval system. Existing tool retrieval methods primarily focus on semantic matching between user queries and tool descriptions, frequently leading to the retrieval of redundant, similar tools. Consequently, these methods fail to provide a complete set of diverse tools necessary for addressing the multifaceted problems encountered by LLMs. In this paper, we propose a novel model-agnostic **CO**llaborative **L**earning-based **T**ool Retrieval approach, **COLT**, which captures not only the semantic similarities between user queries and tool descriptions but also takes into account the collaborative information of tools. Specifically, we first fine-tune the PLM-based retrieval models to capture the semantic relationships between queries and tools in the semantic learning stage. Subsequently, we construct three bipartite graphs among queries, scenes, and tools and introduce a dual-view graph collaborative learning framework to capture the intricate collaborative relationships among tools during the collaborative learning stage. Extensive experiments on both the open benchmark and the newly introduced ToolLens dataset show that COLT achieves superior performance. Notably, the performance of BERT-mini (11M) with our proposed model framework outperforms BERT-large (340M), which has 30 times more parameters. Furthermore, we will release ToolLens publicly to facilitate future research on tool retrieval.

CCS CONCEPTS

• **Information systems → Information retrieval**.

KEYWORDS

Tool Retrieval, Retrieval Completeness, Large Language Model

ACM Reference Format:
Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Towards Completeness-Oriented Tool Retrieval for Large Language Models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679847

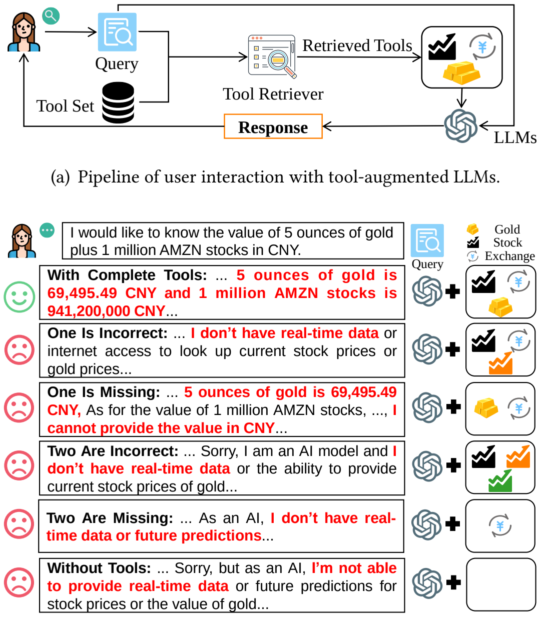
1 INTRODUCTION

Recently, large language models (LLMs) have demonstrated remarkable progress across various natural language processing tasks [1, 2, 4, 41]. However, they often struggle with solving highly complex problems and providing up-to-date knowledge due to the constraints of their pre-training data [23, 42]. A promising approach to overcome these limitations is tool learning [18, 26, 29, 31, 48], which enables LLMs to dynamically interact with external tools, significantly facilitating access to real-time data and the execution of complex computations. By integrating tool learning, LLMs transcend the confines of their outdated or limited pre-trained knowledge [2], offering responses to user queries with significantly improved accuracy and relevance [14, 28]. However, real-world systems typically involve a large number of tools, making it impractical to take the descriptions of all tools as input for LLMs due to length limitations and latency constraints. Thus, as illustrated in Figure 1(a), developing an effective tool retrieval system becomes essential to fully exploit the potential of tool-augmented LLMs [8].

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

CIKM '24, October 21–25, 2024, Boise, ID, USA.

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679847



(a) Pipeline of user interaction with tool-augmented LLMs.

I would like to know the value of 5 ounces of gold plus 1 million AMZN stocks in CNY.

With Complete Tools: ... **5 ounces of gold is 69,495.49 CNY and 1 million AMZN stocks is 941,200,000 CNY**...

One Is Incorrect: ... **I don't have real-time data** or internet access to look up current stock prices or gold prices...

One Is Missing: ... **5 ounces of gold is 69,495.49 CNY,** As for the value of 1 million AMZN stocks, ..., **I cannot provide the value in CNY**...

Two Are Incorrect: ... Sorry, I am an AI model and **I don't have real-time data** or the ability to provide current stock prices of gold...

Two Are Missing: ... As an AI, **I don't have real-time data or future predictions**...

Without Tools: ... Sorry, but as an AI, **I'm not able to provide real-time data** or future predictions for stock prices or the value of gold...

(b) Illustration of different responses with different tools.

Figure 1: An illustration of tool retrieval for LLMs with tool learning and varied responses using different tools.

Typically, existing tool retrieval approaches directly employ dense retrieval techniques [8, 28, 49], solely focusing on matching semantic similarities between queries and tool descriptions. However, these approaches may fall short when addressing multifaceted queries that require a collaborative effort from multiple tools to formulate a comprehensive response. For instance, in Figure 1(b), consider a user's request to calculate the value of 5 ounces of gold plus 1 million AMZN stocks in CNY. Such a query requires the simultaneous use of tools for gold prices, stock values, and currency exchange rates. The absence of any of these tools yields an incomplete answer. In this example, dense retrieval methods that rely solely on semantic matching may retrieve multiple tools related to stock prices while neglecting others. This highlights a significant limitation of dense retrieval methods that overlook the necessity for tools to interact collaboratively. Thus, ensuring the completeness of retrieved tools is an essential aspect of a tool retrieval system, which is often neglected by traditional retrieval approaches.

Toward this end, this paper proposes **COLT**, a novel model-agnostic **CO**llaborative **L**earning-based **T**ool retrieval approach aimed at enhancing completeness-oriented tool retrieval. This method is structured into two main stages: semantic learning and collaborative learning. Initially, we fine-tune traditional pre-trained language models (PLMs) on tool retrieval datasets to acquire semantic matching information between queries and tools, thereby addressing the potential performance issues of these models in zero-shot scenarios for tool retrieval tasks. Subsequently, to capture the intricate collaborative relationship among tools, a concept of "scene" is proposed to indicate a group of collaborative tools. Based on this, COLT integrates three bipartite graphs among queries, scenes, and tools. More specifically, given the initial semantic embedding from the semantic learning stage, the high-order collaborative relationship is better integrated via message propagation and cross-view graph contrastive learning among these graphs. The learning objective incorporates a list-wise multi-label loss to ensure the simultaneous acquisition of tools from the entire ground-truth set without favoring any specific tool.

Moreover, traditional retrieval metrics like Recall [52] and NDCG [16] fail to capture the completeness necessary for effective tool retrieval. As illustrated in Figure 1(b), the exclusion of any essential tool from the ground-truth tool set compromises the ability to fully address user queries, indicating that metrics focused solely on individual tool ranking performance are inadequate when multiple tools are required. To bridge this gap, we introduce COMP@K, a new metric designed to assess tool retrieval performance based on completeness, which can serve as a reliable indicator of how well a tool retrieval system performs for downstream tool learning applications. Additionally, we construct a new dataset called ToolLens, in which a query is typically solved with multiple relevant but diverse tools, reflecting the multifaceted nature of user requests in real-world scenarios.

In summary, our main contributions are as follows:

- The collaborative relationships among multiple tools in LLMs have been thoroughly studied, which reveals that incomplete tool retrieval hinders accurate answers, underscoring the integral role each tool plays in the collective functionality.
- We introduce COLT, a novel tool retrieval approach that uses message propagation and cross-view graph contrastive learning among queries, scenes, and tools, incorporating better collaborative information among various tools.
- Extensive experiments demonstrate the superior performance of COLT against state-of-the-art dense retrieval methods in both tool retrieval and downstream tool learning.
- We introduce a new dataset and a novel evaluation metric specifically designed for assessing multi-tool usage in LLMs, which will facilitate future research on tool retrieval.

2 RELATED WORK

Tool Learning. Recent studies highlight the potential of LLMs to utilize tools in addressing complex problems [24, 27]. Existing tool learning approaches can be categorized into two types: tuning-free and tuning-based methods [8]. Tuning-free methods capitalize on the in context learning ability of LLMs through strategic prompting [32, 33, 43, 47]. For example, ART [25] constructs a task library, from which it retrieves demonstration examples as few-shot prompts when encountering real-world tasks. Conversely, tuning-based methods involve directly fine-tuning the parameters of LLMs on specific tool datasets to master tool usage. For example, ToolLLaMA [28] employs the instruction-solution pairs derived from the DFSDT method to fine-tune the LLaMA model, thereby significantly enhancing its tool usage capabilities. Despite these advancements, most strategies either provide a manual tool set [31, 38, 46] or employ simple dense retrieval [8] for tool retrieval. However, LLMs must choose several useful tools from a vast array of tools in real-world applications, necessitating a robust tool retriever to address the length limitations and latency constraints of LLMs.

Tool Retrieval. Tool retrieval aims to find top-K most suitable tools for a given query from a vast set of tools. Existing tool retrieval methods typically directly adopt traditional retrieval approaches, and state-of-the-art retrieval methods can be categorized into two types: term-based and semantic-based. Term-based methods, such as TF-IDF [35] and BM25 [30], prioritize term matching via sparse representations. Conversely, semantic-based methods, such as ANCE [45], TAS-B [12], coCondensor [7], and Contriever [15], utilize neural networks to learn the semantic relationship between queries and tool descriptions and then calculate the semantic similarity using methods such as cosine similarity. Despite these advancements, existing methods for tool retrieval overlook the importance of the collaborative relationship among multiple tools, thereby falling short of meeting the completeness criterion for tool retrieval. Our work seeks to mitigate these issues by collaborative learning that leverages graph neural networks and cross-view contrastive learning among graphs.

3 OUR APPROACH: COLT

In this section, we first introduce task formulation of tool retrieval. Then we describe the details of the proposed COLT approach.

3.1 Task Formulation

Formally, given a user query $q \in \mathcal{Q}$, the goal of tool retrieval is to filter out the top-K most suitable tools $\{t^{(1)}, t^{(2)}, \ldots, t^{(K)}\}$ from the entire tool set $\mathcal{T} = \{(t_1, d_1), (t_2, d_2), \ldots, (t_N, d_N)\}$, where each element represents a specific tool t_i associated with its description d_i, and N is the number of tools in the tool set.

Goal. As discussed in Section 1, the comprehensiveness of the tools retrieved is crucial for LLMs to enhance their ability to accurately address multifaceted and real-time questions. Therefore, it is necessary to ensure that the retrieved tools encompass all the tools required by the user question. Considering these factors, the goal of tool retrieval is to optimize both accuracy and completeness, ensuring the provision of desired tools for downstream tasks.

3.2 Overview of COLT

As illustrated in Figure 2, COLT employs a two-stage learning strategy, which includes semantic learning followed by collaborative learning. In the first stage, the semantic learning module processes both queries and tools to derive their semantic representations, aiming to align these representations closely within the semantic space. Subsequently, the collaborative learning module enhances these preliminary representations by introducing three bipartite graphs among queries, scenes, and tools. Through dual-view graph contrastive learning within these three bipartite graphs, COLT is able to capture the high-order collaborative information between tools. Furthermore, a list-wise multi-label loss is utilized in the learning objective to facilitate the balanced retrieval of diverse tools from the complete ground-truth set, avoiding undue emphasis on any specific tool.

In the following sections, we will present the details of these two key learning stages in COLT.

3.3 Semantic Learning

As shown in Figure 2 (a), in the first stage of COLT, we adopt the established dense retrieval (DR) framework [9, 50], leveraging pre-trained language models (PLMs) such as BERT [17] to encode both the query q and the tool t into low dimensional vectors. Specifically, we employ a bi-encoder architecture, with the cosine similarity between the encoded vectors serving as the initial relevance score:

$$\widehat{y}_{\mathrm{SL}}(q, t) = \mathrm{sim}(\mathbf{e}_q, \mathbf{e}_t), \tag{1}$$

where $\mathbf{e}_q$ and $\mathbf{e}_t$ are the mean pooling vectors from the final layer of the PLM, and $\mathrm{sim}(\cdot, \cdot)$ represents the cosine similarity function.

For training, we utilize the InfoNCE loss [10, 45], a standard contrastive learning technique used in training DR models, which contrasts positive pairs against negative ones. Specifically, given a query q, its relevant tool t^+ and the set of irrelevant tools $\{t_1^-, \cdots, t_k^-\}$, we minimize the following loss:

$$-\log \frac{e^{\mathrm{sim}(q,t^+)}}{e^{\mathrm{sim}(q,t^+)} + \sum_{j=1}^{k} e^{\mathrm{sim}(q,t_j^-)}}. \tag{2}$$

Through this loss function, we can increase the similarity score between the query and its relevant tool while decreasing the similarity scores between the query and irrelevant tools.

This semantic learning phase ensures good representations for each query and tool from the text description view. However, relying solely on semantic-based retrieval is insufficient for completeness-oriented tool retrieval, as it often falls short in addressing multifaceted queries effectively.

3.4 Collaborative Learning

3.4.1 Bipartite Graphs in Tool Retrieval. To capture the collaborative information between tools and achieve completeness-oriented tool retrieval, we first formulate the relationship between queries and tools with three bipartite graphs. Specifically, as illustrated in Figure 2 (b), we conceptualize the ground-truth tool set for each query as a "scene", considering that a collaborative operation of multiple tools is essential to fully address multifaceted queries. For example, given the query "I want to travel to Paris.", it doesn't merely seek a single piece of information but initiates a "scene" of travel planning, which involves using various tools for transportation, weather forecasts, accommodation choices, and details about attractions. This scenario underscores the need for scene matching beyond traditional semantic search or recommendation scenarios, where the focus is on selecting any relevant documents or items without considering their collaborative utility. As a result, traditional semantic-based retrieval systems may only retrieve tools related to Paris attractions, thus failing to provide a comprehensive and complete tool set for the LLMs. Conversely, we construct three bipartite graphs linking queries, scenes, and tools, i.e., Q-S (Query-Scene) graph, Q-T (Query-Tool) graph, and S-T (Scene-Tool) graph. By formulating these three graphs, we can further capture the high-order relationships among tools with graph learning, facilitating a scene-based understanding that aligns to achieve completeness-oriented tool retrieval.

3.4.2 Dual-view Graph Collaborative Learning. Leveraging the initial query and tool representations derived from the first-stage semantic learning, along with the three constructed bipartite graphs,

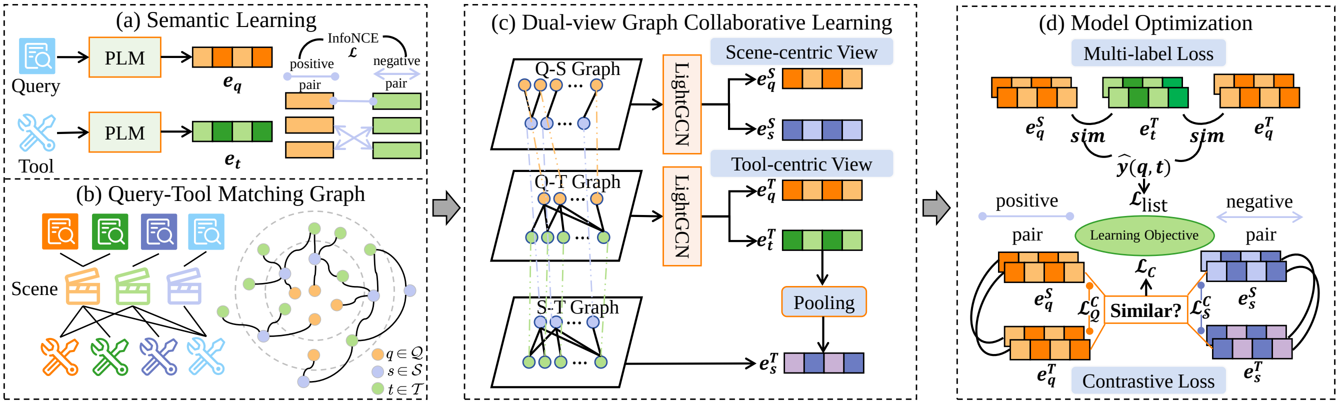


Figure 2: The architecture of the proposed two-stage learning framework COLT for tool retrieval.

we introduce a dual-view graph collaborative learning framework. This framework is designed to capture the relationships between tools, as depicted in Figure 2 (c). It assesses the relevance between queries and tools from two views:

• **Scene-centric View:** Through the Q-S graph and S-T graph, this view captures the relevance between queries and tools mediated by a scene. This offers a nuanced view that considers the collaborative context in which tools work together to meet the requirements of a query.

• **Tool-centric View:** Utilizing the Q-T graph, this view establishes a direct relevance between each query and its corresponding tools, providing a straightforward measure of their relevance.

This dual-view framework allows for comprehensive access to query-tool relevance, integrating both direct relevance and the broader context of tool collaboration within scenes, thereby enhancing the completeness of tool retrieval.

For the scene-centric view, we adopt the simple but effective Graph Neural Network (GNN)-based LightGCN [11] model to delve into the complex relationships between queries and scenes. This is achieved through iterative aggregation of neighboring information across I layers within the Q-S graph. The aggregation process for the i-th layer, enhancing the representations of queries $\mathbf{e}_q^{S(i)}$ and scenes $\mathbf{e}_s^{S(i)}$, is defined as follows:

$$\begin{cases} \mathbf{e}_q^{S(i)} = \sum\limits_{s \in \mathcal{N}_q^S} \dfrac{1}{\sqrt{|\mathcal{N}_q^S|}\sqrt{|\mathcal{N}_s^Q|}} \mathbf{e}_s^{S(i-1)}, \\ \mathbf{e}_s^{S(i)} = \sum\limits_{q \in \mathcal{N}_s^Q} \dfrac{1}{\sqrt{|\mathcal{N}_q^S|}\sqrt{|\mathcal{N}_s^Q|}} \mathbf{e}_q^{S(i-1)}, \end{cases} \tag{3}$$

where $\mathcal{N}_q^S$, $\mathcal{N}_s^Q$ represent the sets of neighbors of query q and scene s in the Q-S graph, respectively. $\mathbf{e}_q^{S(0)}$ originates from the representations acquired in the first semantic learning stage, while $\mathbf{e}_s^{S(0)}$ is derived from the mean pooling of the representations of ground-truth tools associated with each scene:

$$\mathbf{e}_s^{S(0)} = \frac{1}{|\mathcal{N}_s^T|} \sum_{t \in \mathcal{N}_s^T} \mathbf{e}_t, \tag{4}$$

where $\mathcal{N}_s^T$ represents the set of first-order neighbors of scene s in the S-T graph.

Then we sum the representations from the 0-th layer to the I-th layer to get the final query representations $\mathbf{e}_q^S$ and scene representation $\mathbf{e}_s^S$ for the scene-centric view:

$$\begin{cases} \mathbf{e}_q^S = \mathbf{e}_q^{S(0)} + \cdots + \mathbf{e}_q^{S(I)}, \\ \mathbf{e}_s^S = \mathbf{e}_s^{S(0)} + \cdots + \mathbf{e}_s^{S(I)}. \end{cases} \tag{5}$$

In parallel with the scene-centric view, the tool-centric view utilizes LightGCN on the Q-T graph to refine query and tool representations through iterative aggregation. For each layer i, the enhanced representations, $\mathbf{e}_q^{T(i)}$ for queries and $\mathbf{e}_t^{T(i)}$ for tools, are derived as follows:

$$\begin{cases} \mathbf{e}_q^{T(i)} = \sum\limits_{t \in \mathcal{N}_q^T} \dfrac{1}{\sqrt{|\mathcal{N}_q^T|}\sqrt{|\mathcal{N}_t^Q|}} \mathbf{e}_t^{T(i-1)}, \\ \mathbf{e}_t^{T(i)} = \sum\limits_{q \in \mathcal{N}_t^Q} \dfrac{1}{\sqrt{|\mathcal{N}_q^T|}\sqrt{|\mathcal{N}_t^Q|}} \mathbf{e}_q^{T(i-1)}, \end{cases} \tag{6}$$

where $\mathcal{N}_q^T$, $\mathcal{N}_t^Q$ represent the first-order neighbors of query q and tool t in the Q-T graph, respectively. $\mathbf{e}_q^{T(0)}$ and $\mathbf{e}_t^{T(0)}$ are obtained from the first semantic learning stage.

Then we sum the representations from the 0-th layer to the I-th layer to derive the final query representations $\mathbf{e}_q^T$ and tool representation $\mathbf{e}_t^T$ for the tool-centric view:

$$\begin{cases} \mathbf{e}_q^T = \mathbf{e}_q^{T(0)} + \cdots + \mathbf{e}_q^{T(I)}, \\ \mathbf{e}_t^T = \mathbf{e}_t^{T(0)} + \cdots + \mathbf{e}_t^{T(I)}. \end{cases} \tag{7}$$

Furthermore, leveraging the learned tool representations $\mathbf{e}_t^T$ and the S-T graph, the scene representation $\mathbf{e}_s^T$ within the tool-centric view can be obtained by pooling all related tool representations:

$$\mathbf{e}_s^T = \frac{1}{|\mathcal{N}_s^T|} \sum_{t \in \mathcal{N}_s^T} \mathbf{e}_t^T. \tag{8}$$

Algorithm 1 The Learning Algorithm of COLT

```
Input: PLM, semantic learning training epoch $E$, Query-scene bipartite graph, query-tool bipartite graph, scene-tool bipartite graph, learning rate $lr$, weight decay, layer number $I$, contrastive loss weight $\lambda$, temperature coefficient $\tau$, list length $L$.
Output: COLT Model with learnable parameters $\theta$.
// Semantic Learning:
for $e = 1$ to $E$ do
    Calculate the InfoNCE loss using Eq. (2)
    Update parameter of PLM using AdaW
end for
// Collaborative Learning:
Calculate initial $\mathbf{e}_q^{S(0)}$, $\mathbf{e}_s^{S(0)}$, $\mathbf{e}_q^{T(0)}$ and $\mathbf{e}_t^{T(0)}$ using the embeddings obtained from the first-stage semantic learning and Eq. (4)
while COLT not Convergence do
    for $i = 1$ to $I$ do
        Conduct message propagation using Eq. (3) and Eq. (6)
    end for
    Calculate final $\mathbf{e}_q^S$, $\mathbf{e}_s^S$, $\mathbf{e}_q^T$, $\mathbf{e}_s^T$ and $\mathbf{e}_t^T$ using Eq. (5), Eq. (7) and Eq. (8)
    Calculate contrastive loss $\mathcal{L}_Q^C$ and $\mathcal{L}_S^C$ using Eq. (10) and Eq. (11)
    Calculate multi-label loss $\mathcal{L}_{\text{list}}$ using Eq. (14)
    Calculate total loss $\mathcal{L}$ using Eq. (15)
    Update model parameter using Adam
end while
return $\theta$
```

In summary, our dual-view graph collaborative learning framework yields two sets of embeddings: $\mathbf{e}_q^S$ and $\mathbf{e}_s^S$ from the scene-centric view, and $\mathbf{e}_q^T$ and $\mathbf{e}_s^T$ from the tool-centric view for queries and scenes, respectively. Then, the final matching score of each query-tool pair (q, t) is calculated using the following formula:

$$\widehat{y}(q,t) = \text{sim}(\mathbf{e}_q^S, \mathbf{e}_t^T) + \text{sim}(\mathbf{e}_q^T, \mathbf{e}_t^T). \tag{9}$$

3.4.3 Learning Objective. As shown in Figure 2 (d), we capture high-order collaborative relationships between tools and align the cooperative interactions across two views using a cross-view contrastive loss. Specifically, the representations of queries and scenes can be learned by optimizing the cross-view InfoNCE [10, 37] loss:

$$\mathcal{L}_Q^C = -\frac{1}{|Q|}\sum_{q\in Q}\log\frac{e^{\text{sim}(\mathbf{e}_q^S,\mathbf{e}_q^T)/\tau}}{\sum_{q-\in Q} e^{\text{sim}(\mathbf{e}_q^S,\mathbf{e}_{q-}^T)/\tau}}, \tag{10}$$

$$\mathcal{L}_S^C = -\frac{1}{|\mathcal{S}|}\sum_{s\in \mathcal{S}}\log\frac{e^{\text{sim}(\mathbf{e}_s^S,\mathbf{e}_s^T)/\tau}}{\sum_{s-\in \mathcal{S}} e^{\text{sim}(\mathbf{e}_s^S,\mathbf{e}_{s-}^T)/\tau}}, \tag{11}$$

where τ is the temperature parameter.

To ensure the complete retrieval of diverse tools from the full set of ground-truth tools, without favoring any particular tool, we design a list-wise multi-label loss as the main learning objective loss. Given a query q, the labeled training data is $\Gamma_q = \{\mathcal{T}_q = \{t_i, d_i\}, y = \{y(q,t_i)\} | 1 \leq i \leq L\}$, where $\mathcal{T}_q$ denotes a tool list with length L, comprising N_q ground-truth tools and $L - N_q$ negative tools that are randomly sampled from the entire tool set. $y(q, t_i)$ is the binary relevance label, taking a value of either 0 or 1, and the ideal scoring function should meet the following criteria:

$$p_q^t = \frac{\gamma(y(q,t))}{\sum_{t'\in\mathcal{T}_q}\gamma(y(q,t'))}, \tag{12}$$

Table 1: Statistics of the experimental datasets. Tools/Query denotes the number of ground-truth tools for each query.

Dataset	# Query			# Tool	# Tools/Query
	Training	Testing	Total		
ToolLens	16,893	1,877	18,770	464	1 ∼ 3
ToolBench (I2)	74,257	8,250	82,507	11,473	2 ∼ 4
ToolBench (I3)	21,361	2,373	23,734	1,419	2 ∼ 4

where p_q^t is the probability of selecting tool t. $\gamma(y(q,t)) = 1$ if $y(q,t) = 1$ and $\gamma(y(q,t)) = 0$ if $y(q,t) = 0$.

Similarly, given the predicted scores $\{\widehat{y}(q,t_1), \cdots, \widehat{y}(q,t_L)\}$, the probability of selecting tool t can be derived:

$$\widehat{p_q^t} = \frac{\gamma(\widehat{y}(q,t))}{\sum_{t'\in\mathcal{T}_q}\gamma(\widehat{y}(q,t'))}. \tag{13}$$

Therefore, the list-wise multi-label loss function minimizes the discrepancy between these two probability distributions:

$$\mathcal{L}_{\text{list}} = -\sum_{q\in Q}\sum_{t\in\mathcal{T}_q} p_q^t \log\widehat{p_q^t} + (1-p_q^t)\log(1-\widehat{p_q^t}), \tag{14}$$

Based on the multi-label loss $\mathcal{L}_{\text{list}}$ and the contrastive loss $\mathcal{L}_Q^C$, the final loss $\mathcal{L}$ for our proposed COLT is formally defined as:

$$\mathcal{L} = \mathcal{L}_{\text{list}} + \lambda(\mathcal{L}_Q^C + \mathcal{L}_S^C), \tag{15}$$

where λ is the co-efficient to balance the two losses.

The learning algorithm of COLT is summarized in Algorithm 1.

4 DATASETS

To verify the effectiveness of COLT, we utilize two datasets for multi-tool scenarios: ToolBench and a newly constructed dataset, ToolLens. We randomly select 10% of the entire dataset to serve as the test data. The statistics of the datasets after preprocessing are summarized in Table 1.

ToolBench. ToolBench [28] is a benchmark commonly used to evaluate the capability of LLMs in tool usage. In our experiments, we notice that its three subsets exhibit distinct characteristics. The first subset (I1) focuses on single-tool scenarios, which diverges from our emphasis on multi-tool tasks. However, both the second subset (I2) and the third subset (I3) align with our focus on multi-tool tasks. Therefore, we chose I2 and I3 as the primary datasets for our experiments.

ToolLens. While existing datasets like ToolBench [28] and TOOLE [14] provide multi-tool scenarios, they present limitations. TOOLE encompasses only 497 queries, and ToolBench’s dataset construction, which involves providing complete tool descriptions to ChatGPT, results in verbose and semantically direct queries. These do not accurately reflect the brief and often multifaceted nature of real-world user queries. To address these shortcomings, we introduce ToolLens, crafted specifically for multi-tool scenarios.

As shown in Figure 3, the creation of ToolLens involves a novel five-step methodology: **1) Tool Selection:** To create a high-quality tool dataset, we rigorously filter ToolBench, focusing on 464 available and directly callable tools relevant to everyday user queries, excluding those for authentication or testing. **2) Scene Mining:**

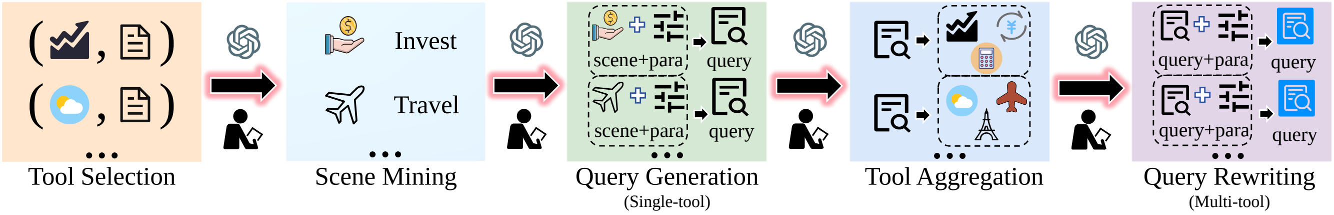


Figure 3: An overview of the dataset construction pipeline of ToolLens. Human verification is included at each step.

Table 2: Quality verification of ToolLens.

Evaluator	ToolLens vs. ToolBench			ToolLens vs. TOOLE		
	Whether the query is natural?					
GPT-4	ToolLens 68%	ToolBench 14%	Equal 18%	ToolLens 44%	TOOLE 36%	Equal 20%
Human	ToolLens 64%	ToolBench 10%	Equal 26%	ToolLens 54%	TOOLE 24%	Equal 22%
	Whether the user intent is multifaceted?					
GPT-4	ToolLens 62%	ToolBench 14%	Equal 24%	ToolLens 50%	TOOLE 24%	Equal 26%
Human	ToolLens 60%	ToolBench 12%	Equal 28%	ToolLens 58%	TOOLE 18%	Equal 24%

We prompt GPT-4 to generate potential scenes that are relevant to the detailed descriptions of the selected tools, and ensure their validity through human verification. **3) Query Generation:** We then employ GPT-4 to generate queries based on the provided scene and the parameters required for tool calling. Notably, we avoid providing the complete tool description to GPT-4 to avoid the generated query being closely aligned with the tool. **4) Tool Aggregation:** The queries generated in the aforementioned way are only relevant to a single tool. To enhance the relevance of queries across multiple tools, we reprocess them through GPT-4 to identify categories of tools that could be relevant, which are then aligned with our tool set through dense retrieval and manual verification. **5) Query Rewriting:** Finally, we utilize GPT-4 to revise queries to incorporate all necessary parameters by providing it with both the initial query and a list of required parameters, thereby yielding concise yet intentionally multifaceted queries that better mimic real-world user interactions. It is worth noting that we incorporate a human verification process at each step to ensure data quality.

This comprehensive construction pipeline ensures ToolLens accurately simulates real-world tool retrieval dynamics. The resulting ToolLens dataset includes 18,770 queries and 464 tools, with each query being associated with 1 ∼ 3 verified tools.

Discussion and Quality Verification. Unlike prior datasets, ToolLens uniquely focuses on creating natural, concise, and multifaceted queries to reflect real-world demands. To assess the quality of ToolLens, following previous works [8, 21, 34], we employ GPT-4 as an evaluator and human evaluation where three well-educated doctor students are invited to evaluate 50 randomly sampled cases from ToolLens, ToolBench and TOOLE in the following two aspects:(1) Natural-query: whether the query is natural. (2) Multifaceted intentions: whether the user intent is multifaceted. The results are illustrated in Table 2. In most cases, ToolLens outperforms ToolBench and TOOLE. Furthermore, using GPT-4 as the evaluator shows a high degree of consistency with human evaluation trends, which underscores the validity of employing GPT-4 as an evaluator.

5 EXPERIMENTS

In this section, we first describe the experimental setups and then conduct an extensive evaluation and analysis of the proposed COLT. The source code and the proposed ToolLens dataset are publicly available at https://github.com/quchangle1/COLT.

5.1 Experimental Setups

5.1.1 Evaluation Metrics. Following the previous works [8, 28, 51], we utilize the widely used retrieval metrics Recall@K and NDCG@K and report the metrics for $K \in \{3, 5\}$. However, as discussed in Section 1, Recall and NDCG do not adequately fulfill the requirements of completeness that are crucial for effective tool retrieval. To further tailor our assessment to the specific challenges of tool retrieval tasks, we also introduce a new metric, COMP@K. This metric is designed to measure whether the top-K retrieved tools form a complete set with respect to the ground-truth set:

$$\text{COMP}@K = \frac{1}{|Q|}\sum_{q=1}^{|Q|} \mathbb{I}(\Phi_q \subseteq \Psi_q^K),$$

where Φ_q is the set of ground-truth tools for query q, Ψ_q^K represents the top-K tools retrieved for query q, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the retrieval results include all ground-truth tools within the top-K results for query q, and 0 otherwise.

5.1.2 Baselines. As our proposed COLT is model-agnostic, we apply it to several representative PLM-based retrieval models (as backbone models) to validate the effectiveness:

ANCE[45] uses a dual-encoder architecture with an asynchronously updated ANN index for training, enabling global selection of hard negatives. **TAS-B**[12] is a bi-encoder that employs balanced margin sampling to ensure efficient query sampling from clusters per batch. **co-Condenser**[7] uses a query-agnostic contrastive loss to cluster related text segments and distinguish unrelated ones. **Contriever**[15] leverages inverse cloze tasks, cropping for positive pair generation, and momentum contrastive learning to achieve state-of-the-art zero-shot retrieval performance.

In addition to PLM-based dense retrieval methods, we also compare with the classical lexical retrieval model BM25, widely used for tool retrieval as documented in [8, 28]. **BM25** [30] uses an inverted index to identify suitable tools based on exact term matching.

Table 3: Performance comparison of different tool retrieval methods on ToolLens and ToolBench datasets. "†" denotes the best results for each column. The term "Zero-shot" refers to the performance of dense retrieval models without any training. "+Fine-tune" indicates that retrieval models are fine-tuned on ToolLens and ToolBench datasets. "+COLT (Ours)" indicates that dense retrieval backbones are equipped with our proposed method. R@K, N@K, and C@K are short for Recall@K, NDCG@K and COMP@K, respectively.

Backbone	Framework	ToolLens						ToolBench (I2)						ToolBench (I3)					
		R@3	R@5	N@3	N@5	C@3	C@5	R@3	R@5	N@3	N@5	C@3	C@5	R@3	R@5	N@3	N@5	C@3	C@5
BM25	-	21.58	26.88	23.19	26.09	3.89	6.13	17.06	21.38	17.83	19.88	2.39	4.37	29.33	35.88	32.20	35.08	5.52	9.78
ANCE	Zero-shot	20.82	26.56	21.45	24.57	5.06	7.46	20.82	26.56	21.45	24.57	5.06	7.46	21.55	26.38	23.44	25.60	2.44	4.59
	+Fine-tune	80.62	94.17	82.35	90.15	54.23	85.83	58.58	67.20	58.58	63.75	26.46	42.80	65.11	76.63	69.27	74.14	34.68	53.64
	+COLT (Ours)	**92.15**	**97.78**†	**92.78**	**96.10**	**80.50**	**94.40**	**70.76**	**80.59**	**73.64**	**77.98**	**45.10**	**62.93**	**73.37**	**83.97**	**77.95**	**82.14**	**46.01**	**66.41**
TAS-B	Zero-shot	19.10	23.71	19.81	22.33	5.17	7.14	19.10	23.71	19.81	22.33	5.17	7.14	25.32	31.15	27.80	30.36	3.84	6.40
	+Fine-tune	81.26	94.06	82.54	89.94	54.66	85.72	62.78	67.49	58.96	64.21	26.74	43.66	66.04	77.64	70.41	75.34	35.69	55.75
	+COLT (Ours)	**91.49**	**96.91**	**92.48**	**95.63**	**79.00**	**92.22**	**71.64**	**81.12**	**74.60**	**78.74**	**46.77**	**64.38**	**74.49**	**84.58**	**79.03**	**82.95**	**48.16**	**68.35**
coCondensor	Zero-shot	15.33	19.37	16.15	18.32	3.02	5.33	15.33	19.37	16.15	18.32	3.02	5.30	20.80	25.24	23.21	25.10	2.07	3.75
	+Fine-tune	82.37	94.69	83.90	91.06	56.37	86.73	57.70	69.46	60.80	66.07	28.78	46.06	66.97	79.30	71.20	76.50	37.08	58.66
	+COLT (Ours)	**92.65**	**97.78**†	**93.16**	**96.17**	**82.25**	**94.56**†	**73.91**	**83.47**	**76.75**	**80.87**	**49.15**	**67.75**	**75.48**	**84.97**	**80.00**	**83.55**	**49.17**	**68.64**†
Contriever	Zero-shot	25.67	31.15	26.96	29.95	7.46	9.80	25.67	31.15	26.96	29.95	7.46	9.80	31.37	38.60	34.13	37.37	6.03	11.42
	+Fine-tune	83.58	95.17	84.98	91.69	59.46	88.65	58.89	70.75	62.11	67.42	29.77	48.31	68.58	80.05	72.86	77.69	39.70	60.89
	+COLT (Ours)	**93.64**†	**97.75**	**94.53**†	**96.91**†	**84.55**†	**94.08**	**75.72**†	**85.03**†	**78.57**†	**82.54**†	**51.97**†	**70.10**†	**76.63**†	**85.50**†	**81.21**†	**84.18**†	**52.00**†	**68.47**

5.1.3 Implementation Details. We utilize the BEIR [40] framework for dense retrieval baselines, setting the training epochs to 5 with the learning rate of $2e{-}5$, weight decay of 0.01, and using the AdamW optimizer. Our model-agnostic approach directly applies dense retrieval for semantic learning. During collaborative learning, we set the batch size as 2048 and carefully tune the learning rate among $\{1e{-}3, 5e{-}3, 1e{-}4, 5e{-}4, 1e{-}5\}$, the weight decay among $\{1e{-}5, 1e{-}6, 1e{-}7\}$, as well as the layer number I among $\{1, 2, 3\}$.

5.2 Experimental Results

5.2.1 Retrieval Performance. Table 3 presents the results of different tool retrieval methods on ToolLens, ToolBench (I2 and I3). From the results, we have the following observations and conclusions:

We can observe that traditional dense retrieval models perform poorly in zero-shot scenarios, even inferior to that of BM25. This indicates that these models may not be well-suited for tool retrieval tasks. Conversely, the BM25 model significantly lags behind fine-tuned PLM-based dense retrieval methods, underscoring the superior capability of the latter in leveraging contextual information for more effective tool retrieval. Despite this advantage, PLM-based methods fall short in the COMP metric, which is specifically designed for evaluating completeness in tool retrieval scenarios. This suggests that while effective for general retrieval tasks, PLM-based methods may not fully meet the unique demands of tool retrieval.

All base models equipped with COLT exhibit significant performance gains across all metrics on all three datasets, particularly in the COMP@3 metric. These improvements demonstrate the effectiveness of COLT, which can be attributed to the fact that COLT adopts a two-stage learning framework with semantic learning followed by collaborative learning. In this way, COLT can capture the intricate collaborative relationships between tools, resulting in effectively retrieving a complete tool set.

5.2.2 Downstream Tool Learning Performance. To verify that improvements of COLT in tool retrieval truly enhance downstream tool learning applications, we conduct a validation study using the pairwise comparison method [5, 19, 36]. We randomly select 100 queries from the test set of ToolLens and use various retrieval models to retrieve the top-3 tools for each query. Then we utilize GPT-4 as an evaluator, examining the responses generated with different retrieved tools across four dimensions: Coherence, Relevance, Comprehensiveness, and Overall. Specifically, the user query and a pair of responses are utilized as prompts to guide GPT-4 in determining the superior response. Additionally, we also consider that LLMs may respond differently to the order in which text is presented in the prompt [13, 20, 22, 39]. So each comparison is conducted twice with reversed response order to mitigate potential biases from text order, ensuring a more reliable assessment.

Table 4: Elo ratings for different models w.r.t. "Coherence", "Relevance", "Comprehensiveness" and "Overall" evaluated by GPT-4 on ToolLens dataset.

	Evaluation Aspects			
	Coherence	Relevance	Comprehensiveness	Overall
BM25	848	845	860	780
ANCE	934	936	946	1016
TAS-B	995	991	988	1028
coCondensor	1031	1036	1041	1035
Contriever	1076	1082	1044	1046
COLT (Ours)	1116	1110	1121	1096

We establish a tournament-style competition using the Elo ratings system, which is widely employed in chess and other two-player games to measure the relative skill levels of the players [6, 44]. Following previous works [3], we start with a score of 1,000 and set K-factor to 32. Additionally, to minimize the impact of match sequences on Elo scores, we conduct these computations 10,000 times using various random seeds to control for ordering effects.

The results in Table 4 show that superior tool retrieval models can significantly improve downstream tool learning performance. Moreover, responses generated with the tools retrieved from COLT notably outperform those from other methods, achieving the highest Elo ratings in all four assessed dimensions. These results highlight the pivotal role of effective tool retrieval in tool learning applications and further confirm the superiority of COLT.

Table 5: Ablation study of the proposed COLT.

Methods	ToolLens		ToolBench	
	R@|N|	C@|N|	R@|N|	C@|N|
ANCE+COLT (Ours)	**91.08**	**78.36**	**72.22**	**44.28**
w/o semantic learning	36.49	6.84	21.92	1.60
w/o collaborative learning	77.36	49.01	62.39	30.12
w/o list-wise learing	79.94	52.68	66.02	35.82
w/o contrastive learning	85.63	63.87	66.57	34.55
TAS-B+COLT (Ours)	**90.29**	**77.73**	**72.84**	**45.46**
w/o semantic learning	38.49	9.16	32.16	5.47
w/o collaborative learning	76.86	47.83	63.61	31.73
w/o list-wise learning	79.89	52.25	66.91	37.27
w/o contrastive learning	84.86	62.65	67.66	36.36
coCondensor+COLT (Ours)	**91.49**	**79.86**	**74.00**	**47.49**
w/o semantic learning	30.38	5.54	25.07	2.27
w/o collaborative learning	78.83	50.61	64.38	33.08
w/o list-wise learning	81.42	54.16	69.18	40.67
w/o contrastive learning	86.78	67.07	68.92	37.80
Contriever+COLT (Ours)	**92.76**	**82.95**	**75.40**	**49.81**
w/o semantic learning	65.21	30.90	53.33	19.63
w/o collaborative learning	80.60	54.44	68.20	36.91
w/o list-wise learning	81.49	54.93	71.80	46.07
w/o contrastive learning	84.58	60.52	69.46	39.02

5.3 Further Analysis

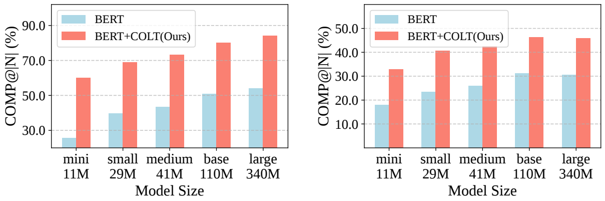
Next, we delve into investigating the effectiveness of COLT. We report the experimental results on the ToolLens and ToolBench (I3) datasets, observing similar trends on ToolBench (I2). Recall@|N| and COMP@|N| are adopted as evaluation metrics, with |N| representing the count of ground-truth tools suitable for each query.

5.3.1 Ablation Study. We conduct ablation studies to assess the impact of various components within our COLT. The results presented in Table 5, highlight the significance of each element:

***w/o* semantic learning** denotes an off-the-shelf PLM is directly employed to get the initial representation for the subsequent collaborative learning stage without semantic learning on the given dataset in Section 3.3. The absence of semantic learning significantly diminishes performance, confirming its essential role in aligning the representations of tools and queries as the basic for the following collaborative learning. Notably, the omission of semantic learning elements markedly reduces performance across other models more than with Contriever. This highlights the superior ability of Contriever in zero-shot learning scenarios compared to the other models.

***w/o* collaborative learning** is a variant where the collaborative learning state is omitted (*i.e.,* only semantic learning). The significant decline in performance in this variant further supports the effectiveness of COLT in capturing the high-order relationships between tools through graph collaborative learning, thereby achieving comprehensive tool retrieval.

***w/o* list-wise learning** refers to a variant that optimizes using pair-wise loss in place of the list-wise loss defined in Eq. (14). This substitution results in a significant drop in performance, highlighting that compared to pairwise loss, list-wise loss optimizes the tools in the same scenario as a whole entity, proving more effective in focusing on completeness.



(a) ToolLens (b) ToolBench

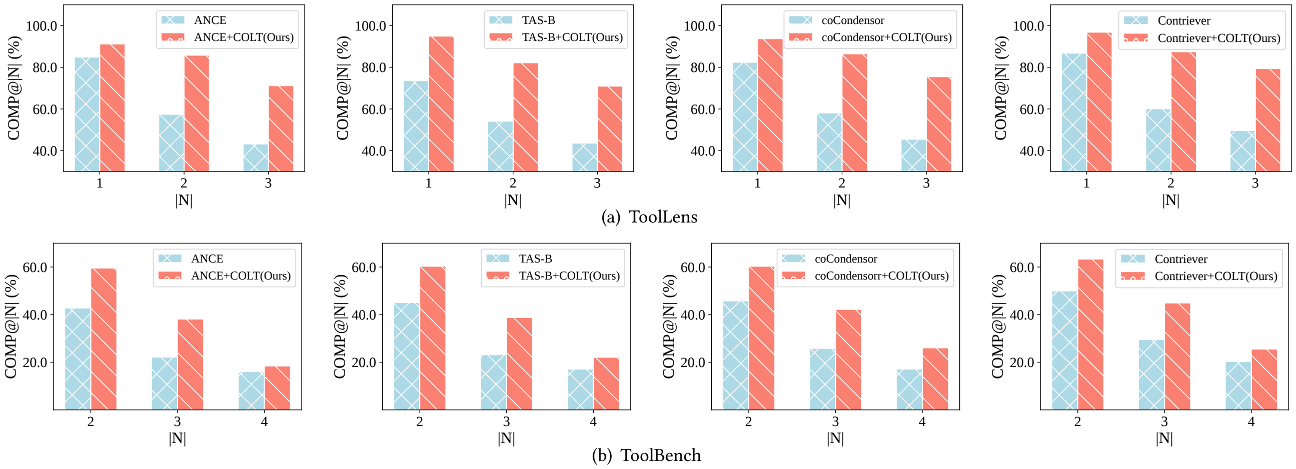
Figure 4: Comparison of different model sizes of PLM.

***w/o* contrastive learning** refers to a variant that optimizes without the contrastive loss defined in Eq. (10) and (11); This omission also leads to a noticeable performance drop, emphasizing the benefits of introducing contrastive learning to achieve better representation for queries and tools within a dual-view learning framework. Additionally, our analysis reveals that contrastive learning is particularly crucial for Contriever, as its absence results in performance lagging behind the other models. This also indicates that the importance of contrastive learning varies across different backbones.

5.3.2 Performance w.r.t. Model Size of PLM. To verify the adaptability and effectiveness of COLT across varying sizes of PLMs, we explore its integration with a range of BERT models, from BERT-mini to BERT-large. This analysis aims to determine whether COLT could generally enhance tool retrieval performance across different model sizes. Figure 4 shows that while the performance of the base model naturally improves with larger PLM sizes, the integration of COLT consistently boosts performance across all sizes. Remarkably, even BERT-mini equipped with COLT, significantly outperforms a much larger BERT-large model (30x larger) operating without our COLT. These results underscore the generalization and robustness of COLT, demonstrating its potential to significantly improve tool retrieval performance for PLMs of any scale.

5.3.3 Performance w.r.t. Different Tool Sizes. The ToolLens dataset encompasses queries that require $1 \sim 3$ tools, while ToolBench includes queries needing $2 \sim 4$ tools. To assess how well COLT adapts to queries with diverse tool requirements, we divide each dataset into three subsets based on the number of tools required by each query and conduct a focused analysis on these subsets. As shown in Figure 5, there is a discernible decline in performance as the number of ground-truth tools increases, reflecting the escalating difficulty of achieving complete retrieval. However, COLT demonstrates consistent performance improvement across all subsets and backbones. This improvement is especially significant in the most challenging cases, where queries may involve using three or four tools. These results consistently highlight the robustness of COLT and its potential to meet the complex demands of tool retrieval tasks across various scenarios.

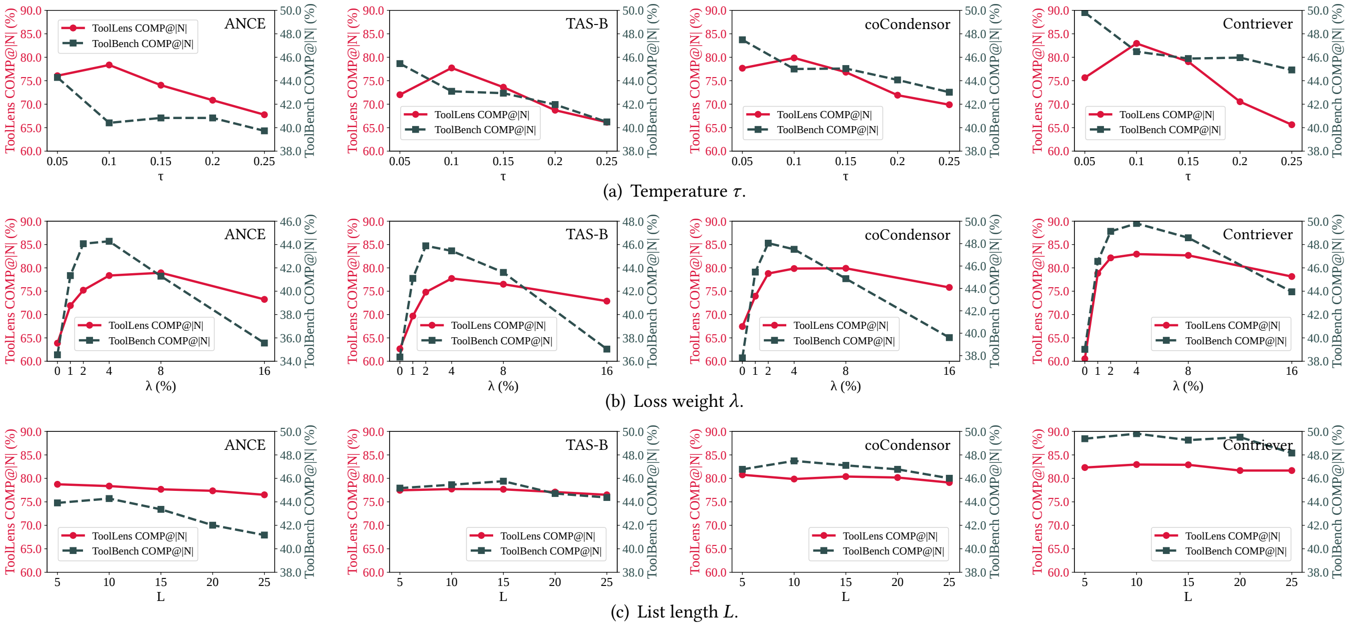
5.3.4 Hyper-parameter Analysis. Figure 6 illustrates the sensitivity of COLT to the temperature parameter τ and the loss weight λ, but shows relative insensitivity to variations in the sampled list length L. The influence of τ varies across two datasets, suggesting that its impact depends on the specific data distribution. Conversely, the pattern observed for λ across both datasets is consistent, marked



(a) ToolLens

(b) ToolBench

Figure 5: Performance comparison regarding different sizes of ground-truth tool sets.



(a) Temperature τ.

(b) Loss weight λ.

(c) List length L.

Figure 6: Sensitivity analysis of COLT performance to hyper-parameters. (a) shows the dependency of model performance on temperature τ. (b) illustrates the influence of loss weight λ. (c) examines the effect of list length L.

by an initial performance improvement that eventually plateaus, underscoring the importance of carefully selecting λ to maximize the effectiveness of COLT.

6 CONCLUSION

This study introduces COLT, a novel model-agnostic approach designed to enhance the completeness of tool retrieval tasks, comprising two stages: semantic learning and collaborative learning. We initially employ semantic learning to ensure semantic representation between queries and tools. Subsequently, by incorporating graph collaborative learning and cross-view contrastive learning, COLT captures the collaborative relationships among tools. Extensive experimental results and analysis demonstrate the effectiveness of COLT, especially in handling multifaceted queries with multiple tool requirements. Furthermore, we release a new dataset ToolLens and introduce a novel evaluation metric COMP, both of which are valuable resources for facilitating future research on tool retrieval.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62377044), Beijing Key Laboratory of Big Data Management and Analysis Methods, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, funds for building world-class universities (disciplines) of Renmin University of China, and PCC@RUC.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

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

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.

[5] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. ACM. https://doi.org/10.1145/3604915.3610646

[6] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs.LG]

[7] Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540* (2021).

[8] Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2024. Confucius: Iterative Tool Learning from Introspection Feedback by Easy-to-Difficult Curriculum. In *AAAI*.

[9] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2022), 1–42.

[10] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[12] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[13] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large Language Models are Zero-Shot Rankers for Recommender Systems. arXiv:2305.08845 [cs.IR]

[14] Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2023. MetaTool Benchmark: Deciding Whether to Use Tools and Which to Use. *arXiv preprint arXiv: 2310.03128* (2023).

[15] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[16] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. 2.

[18] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A Comprehensive Benchmark for Tool-Augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3102–3116. https://doi.org/10.18653/v1/2023.emnlp-main.187

[19] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. arXiv:2211.09110 [cs.CL]

[20] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL]

[21] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. arXiv:2303.16634 [cs.CL]

[22] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. arXiv:2104.08786 [cs.CL]

[23] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511* (2022).

[24] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842* (2023).

[25] Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014* (2023).

[26] Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255* (2022).

[27] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, et al. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354* (2023).

[28] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789* (2023).

[29] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Tool Learning with Large Language Models: A Survey. *arXiv preprint arXiv:2405.17935* (2024).

[30] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[31] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761* (2023).

[32] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems* 36 (2024).

[33] Yifan Song, Weimin Xiong, Dawei Zhu, Cheng Li, Ke Wang, Ye Tian, and Sujian Li. 2023. Restgpt: Connecting large language models with real-world applications via restful apis. *arXiv preprint arXiv:2306.06624* (2023).

[34] Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation Metrics in the Era of GPT-4: Reliably Evaluating Large Language Models on Sequence to Sequence Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8776–8788. https://doi.org/10.18653/v1/2023.emnlp-main.543

[35] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.

[36] Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Instruction Distillation Makes Large Language Models Efficient Zero-shot Rankers. arXiv:2311.01555 [cs.IR]

[37] Jiakai Tang, Sunhao Dai, Zexu Sun, Xu Chen, Jun Xu, Wenhui Yu, Lantao Hu, Peng Jiang, and Han Li. 2024. Towards Robust Recommendation via Decision Boundary-aware Graph Contrastive Learning. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2024).

[38] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. 2023. ToolAlpaca: Generalized Tool Learning for Language Models with 3000 Simulated Cases. *arXiv preprint arXiv:2306.05301* (2023).

[39] Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the Middle: Permutation Self-Consistency Improves Listwise Ranking in Large Language Models. arXiv:2310.07712 [cs.CL]

[40] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[41] 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).

[42] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. 2023. Freshllms: Refreshing large language models with search engine augmentation. *arXiv preprint arXiv:2310.03214* (2023).

[43] 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.

[44] Minghao Wu and Alham Fikri Aji. 2023. Style Over Substance: Evaluation Biases for Large Language Models. arXiv:2307.03025 [cs.CL]

[45] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[46] Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. 2023. On the Tool Manipulation Capability of Open-source Large Language Models. *arXiv preprint arXiv:2305.16504* (2023).

[47] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

[48] Junjie Ye, Guanyu Li, Songyang Gao, Caishuang Huang, Yilong Wu, Sixian Li, Xiaoran Fan, Shihan Dou, Qi Zhang, Tao Gui, et al. 2024. Tooleyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. *arXiv preprint arXiv:2401.00741* (2024).

[49] Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Ren Kan, Dongsheng Li, and Deqing Yang. 2024. EASYTOOL: Enhancing LLM-based Agents with Concise Tool Instruction. *arXiv preprint arXiv:2401.06201* (2024).

[50] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2023. Dense Text Retrieval based on Pretrained Language Models: A Survey. *ACM Trans. Inf. Syst.* (dec 2023).

[51] Yuanhang Zheng, Peng Li, Wei Liu, Yang Liu, Jian Luan, and Bin Wang. 2024. ToolRerank: Adaptive and Hierarchy-Aware Reranking for Tool Retrieval. *In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)* (2024).

[52] Mu Zhu. 2004. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo* 2, 30 (2004), 6.