

# DICE: Dynamic In-Context Example Selection in LLM Agents via Efficient Knowledge Transfer

**Ruoyu Wang \***

UNSW Sydney

ruoyu.wang5@unsw.edu.au

**Junda Wu**

UC San Diego

juw069@ucsd.edu

**Yu Xia**

UC San Diego

yux078@ucsd.edu

**Tong Yu**

Adobe Research

tyu@adobe.com

**Ryan A. Rossi**

Adobe Research

ryrossi@adobe.com

**Julian McAuley**

UC San Diego

jmcauley@ucsd.edu

**Lina Yao**

UNSW Sydney, CSIRO's Data 61

lina.yao@data61.csiro.au

## Abstract

Large language model based agents, empowered by in-context learning (ICL), have demonstrated strong capabilities in complex reasoning and tool-use tasks. However, existing works have shown that the effectiveness of ICL is highly sensitive to the choice of demonstrations, with suboptimal examples often leading to unstable or degraded performance. While prior work has explored example selection, including in some agentic or multi-step settings, existing approaches typically rely on heuristics or task-specific designs and lack a general, theoretically grounded criterion for what constitutes an effective demonstration across reasoning steps. Therefore, it is non-trivial to develop a principled, general-purpose method for selecting demonstrations that consistently benefit agent performance. In this paper, we address this challenge with DICE, Dynamic In-Context Example Selection for LLM Agents, a theoretically grounded ICL framework for agentic tasks that selects the most relevant demonstrations at each step of reasoning. Our approach decomposes demonstration knowledge into transferable and non-transferable components through a causal lens, showing how the latter can introduce spurious dependencies that impair generalization. We further propose a stepwise selection criterion with a formal guarantee of improved agent performance. Importantly, DICE is a general, framework-agnostic solution that can be integrated as a plug-in module into existing agentic frameworks without any additional training cost. Extensive experiments across diverse domains demonstrate our method's effectiveness and generality, highlighting the importance of principled, context-aware demo selection for robust and efficient LLM agents.

## 1 Introduction

Large language model based agents, powered by techniques such as chain-of-thought prompting [48] and agentic frameworks, have become popular and effective for complex reasoning and tool-use tasks [42]. A key component of these frameworks is in-context learning (ICL), where a small set of demonstrations is prepended to the prompt so the model can infer the desired behaviour without any

---

\*Corresponding author

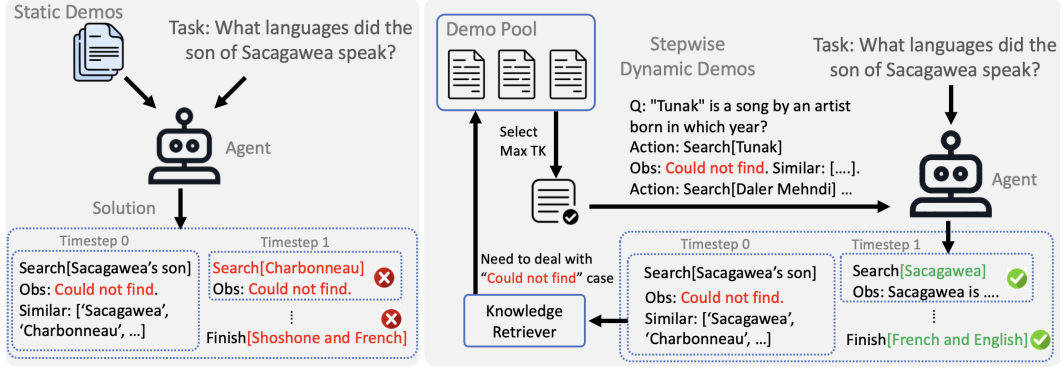


Figure 1: Overview of our stepwise dynamic demonstration selection framework. **Left:** Without relevant demonstrations, the agent struggles when encountering unfamiliar cases—e.g., failing to deal with "Could not find" case - leading to incorrect actions and ultimately a wrong answer. **Right:** With DICE, the agent retrieves contextually relevant examples at each time step by maximizing transferable knowledge (TK). When encountering the "Could not find" case, a relevant demo is selected and guides the agent on handling it, enabling successful completion.

parameter updates [2]. In agentic frameworks such as ReAct [57], demonstrations serve to illustrate how the agent should reason and act based on observations, invoke external tools, and interpret results before proceeding to the next step. By showcasing sequences of thought, action, and observation, ICL guides the agent’s policy and enables effective reasoning in novel environments.

While In-Context Learning (ICL) often achieves strong performance, prior work has shown that its effectiveness is highly sensitive to the choice of examples [59]. To address this instability, some methods have been proposed for active example selection in ICL, aiming to identify more informative or representative exemplars. Some of these approaches focus on one-step question answering tasks [59, 22], while some recent work extended it to multi-step, agentic settings [20, 54]. However, these methods typically rely on heuristics or task-specific strategies, and lack a general, theoretically grounded criterion for what constitutes an effective demonstration across reasoning steps of an agent. This limits their generalizability, particularly in specialized or novel environments where suitable demonstrations may be scarce or hard to identify.

Therefore, it is non-trivial to investigate what constitutes a good demonstration for an LLM-based agent, what knowledge should or should not be transferred from demonstrations to the query task, and to develop a principled, general-purpose method for efficient example selection and knowledge transfer that consistently enhances agent performance across reasoning steps. To address these limitations in the existing literature, we propose DICE, Dynamic In-Context Example Selection, a theoretically grounded ICL framework tailored for agentic tasks. DICE enables agents to dynamically select the most relevant demonstrations at each time step of the problem-solving process, by quantifying the amount of transferable knowledge each demonstration provides (Figure 1). This allows the agent to overcome the limitations of static example selection and to mitigate spurious correlations introduced by irrelevant or misleading demonstrations. In contrast to prior approaches that require additional model training to guide demonstration selection [59, 30, 58], DICE is entirely training-free. It is a general, framework-agnostic solution that can be seamlessly integrated as a plug-in module into existing agentic frameworks without incurring any additional training cost.

Specifically, we begin by formalizing the influence of in-context demonstrations on an agent’s decision-making process through a causal perspective. We decompose the knowledge conveyed by demonstrations into transferable and non-transferable components, demonstrating that the latter can introduce spurious dependencies that hinder generalization. Building on this insight, we propose a dynamic demonstration selection algorithm that, at each step of the problem-solving process, identifies the most relevant examples—those that maximize the transferable knowledge beneficial to the current reasoning step. Crucially, our method is supported by theoretical guarantees, establishing improved bounds on the generalization gap. Our main contributions are as follows:

- We articulate a causal perspective on the empirical instability of ICL, highlighting how demonstrations can degrade performance by introducing spurious association into the

decision-making process. Leveraging this insight, we propose a demonstration selection criterion that mitigates such spurious dependencies by dynamically adapting exemplars to maximize the transferable knowledge relevant to the agent’s subtask at each reasoning step.

- We develop a practical strategy to implement our demonstration selection criterion and instantiate it as a plug-in module compatible with existing agentic frameworks. Our method operates entirely at inference time, requires no additional training, and consistently achieves strong empirical performance across diverse domains and agent architectures.
- We theoretically show that our selection strategy yields tighter generalization bounds by identifying transferable knowledge, leading to improved performance guarantees.

## 2 Method

### 2.1 When Demonstrations Impede Agent Performance?

Intuitively, when using in-context learning (ICL) to guide an agent, the provided examples inevitably contain a mixture of knowledge—some of which is relevant and transferable to the new task, and some of which is not. While the transferable knowledge can be transferred to the query task and help the agent perform better, the irrelevant or task-specific information embedded in the demonstrations may introduce spurious cues. These cues can mislead the agent and ultimately hinder performance on the target task.

We illustrate this intuition using a causal graph, shown in Figure 2, where  $H_t$  represents the agent’s *history* up to the current timestep,  $A_t$  denotes the agent’s next action,  $D$  is the demonstration provided to the agent, and  $TK$  (Transferable Knowledge) refers to the subset of information within  $D$  that is genuinely relevant and beneficial for solving the current decision at step  $t$ . In contrast,  $\epsilon_D$  and  $\epsilon_t$  represent task-specific or spurious knowledge that is irrelevant to other tasks and should ideally not influence decisions beyond their original context. Since  $TK$  captures knowledge shared between the demonstration and the current task, it serves as a common cause of both  $D$  and  $A_t$ , forming the structure  $D \leftarrow TK \rightarrow A_t$ . Additionally,  $H_t \rightarrow A_t$  holds because the agent’s decision at step  $t$  should logically depend on its accumulated experience up to that point. Finally, the task-specific noise terms  $\epsilon_D$  and  $\epsilon_t$  influence only their respective tasks and should not generalize.

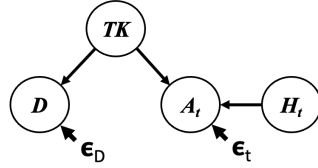


Figure 2: Causal graph showing how demo-specific noise  $\epsilon_D$  affects the next action  $A_t$  in ICL, introducing spurious correlations that can hinder agent performance.

This graph intuitively explains a limitation of standard In-Context Learning (ICL). By providing the demonstration  $D$  as input, we inherently condition on  $D$ , which opens a *collider* structure:  $\epsilon_D \rightarrow D \leftarrow TK \rightarrow A_t$ . As a result, a backdoor path is created, allowing spurious task-specific information from  $\epsilon_D$  to influence the generation of  $A_t$ . This unintended information flow can lead the agent to rely on irrelevant cues, ultimately limiting its performance on the target task.

### 2.2 DICE: Dynamic In-Context Example Selection

To mitigate the limitation described above, we propose a dynamic demonstration selection mechanism that controls the influence of irrelevant knowledge during in-context learning. The core idea is to adaptively select demonstrations most relevant to the agent’s current decision point, thereby reducing the impact of spurious correlations introduced by task-specific information in unrelated examples.

In a standard LLM-based agent framework, an agent interacts with an environment over discrete time steps  $t$ . A *Task* (e.g., a natural-language instruction) and a set of in-context demonstrations *Demos* are provided. At each step  $t$ , the agent takes an action  $a_t \in \mathcal{A}$  and receives an observation  $o_t \in \mathcal{O}$  from the environment, and updates its context  $H_t = (\text{Demos}, \text{Task}, a_1, o_1, \dots, a_t, o_t)$  until the agent emits a special *Finish* action or a time limit  $T$  is reached.

In contrast, our framework render the demonstration set *Demos* dynamic. Specifically, let  $\mathcal{D} = \{d_i\}_{i=1}^N$  with  $d_i = \{\text{Task}_i, a_{i,1}, o_{i,1}, \dots, a_{i,T}, o_{i,T}\}$  be a pool of  $N$  complete trajectories, each illustrating a correct solution to its respective task. At each time step  $t$ , in addition to appending the new action–observation pair  $(a_t, o_t)$  to the context  $H_t$ , we replace the previous *Demos* by selecting a fresh subset of  $M$  trajectories  $\{d_j\}_{j=1}^M \in \mathcal{D}$  via a retrieval policy  $\pi$ , which aims to maximize the

expected cumulative task reward:

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[ \sum_{t=1}^T \mathcal{R}(a_t) \right],$$

where  $\mathcal{R}(a_t)$  measures task-specific performance. By tailoring demonstrations to the agent’s evolving context, our approach filters out irrelevant examples, emphasises transferable knowledge  $TK$ , and boost adaptability while mitigating spurious correlations at each decision step.

To enable such a dynamic ICL framework, an automatic demonstration selection criterion is needed. Therefore, we propose a criterion grounded in the Information Bottleneck (IB) principle, which formalizes the trade-off between preserving useful information and discarding irrelevant signals. We define the objective as follows:

$$\arg \min_{d_i \in D} J_i, \quad \text{where } J_i = I(d_i; \text{TK}_{d_i}) - \beta I(\text{TK}_{d_i}; A_t) \quad (1)$$

where the first term penalizes the inclusion of excessive task-specific details that do not generalize across tasks, and the second term, scaled by a regularization coefficient  $\beta > 0$ , encourages the selection of demonstrations that maximize the mutual information between the agent’s next action  $A_t$  and the transferable knowledge  $TK$ .

This objective naturally arises from our causal analysis in Section 2.1: by maximizing the influence of the causal path  $TK \rightarrow A_t$  while minimizing the effect of the spurious dependency induced by the collider  $\epsilon_D \rightarrow D \leftarrow TK \rightarrow A_t$ , our method selectively filters out non-transferable knowledge. In doing so, it enables more effective and generalizable in-context reasoning, particularly in multi-step tasks where compounding errors from irrelevant information can significantly degrade performance.

### 2.3 Theoretical Guarantee

We conduct a theoretical analysis for the method proposed in Section 2.2. Specifically, we address two key questions: (1) Why is compressing demonstrations to transferable knowledge ( $TK$ ) beneficial? and (2) Why does selecting demonstrations using our proposed criterion improve agent performance? To address these questions, we begin by introducing the notion of the *generalization gap*, a metric that quantifies the discrepancy between a model’s performance on observed examples and its performance on unseen instances.

**Definition 2.1** (Generalization Gap). Let  $(X, Y) \sim \mathcal{D}$  and let  $T = T(X)$  be an encoder output independent of the training data. Given a loss  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$  and predictor  $f : \mathcal{T} \rightarrow \mathcal{Y}$ , the *generalization gap* is:

$$\Delta = \mathbb{E}_{(X, Y) \sim \mathcal{D}} [\ell(f(T), Y)] - \frac{1}{n} \sum_{i=1}^n \ell(f(t_i), y_i),$$

where  $(x_i, y_i)$  are i.i.d. samples from  $\mathcal{D}$  and  $t_i = T(x_i)$ .

This is particularly relevant to our problem, where a demonstration is provided and the agent is expected to solve a different, unseen task. Inspired by existing literature in deep learning theory [52], we have Theorem 2.2. Further details are provided in the Appendix.

**Theorem 2.2.** *Let the encoder  $p(t \mid x)$  be fixed independently of the training data. Then the generalization gap is bounded by:*

$$\Delta \leq \tilde{O} \left( \sqrt{\frac{I(X; T) + 1}{n}} \right),$$

where  $I(X; T)$  is the mutual information between  $X$  and  $T$ , and  $n$  is the number of training samples.

Theorem 2.2 provides a bound on the generalization gap, directly illustrating the benefit of compressing demonstrations to their transferable knowledge. In our setting, let an encoder compress a demonstration  $D$  into its transferable knowledge representation  $TK$ . Then, the generalization gap is bounded by:

$$\Delta(TK) \leq \tilde{O} \left( \sqrt{\frac{I(D; TK) + 1}{n}} \right),$$

In contrast, if no such compression is applied, i.e., we set  $TK = D$ , then  $I(D; TK) = I(D; D) = H(D)$ , yielding the bound:

$$\Delta(D) \leq \tilde{O}\left(\sqrt{\frac{H(D)+1}{n}}\right).$$

Since any non-trivial encoder will discard some non-essential information, we have  $I(D; TK) \leq H(D)$ , which implies  $\Delta(TK) \leq \Delta(D)$ . That means, as soon as  $TK$  omits even a small amount of information from  $D$ , the generalization gap bound becomes strictly tighter. In other words, compressing demonstrations to their transferable components yields a provably stronger high-probability guarantee on the model’s performance.

Furthermore, we show that selecting demonstrations according to our proposed criterion (Equation 1) leads to a tighter generalization bound. To support this claim, we build our analysis on Theorem 2.3, drawing on the theoretical framework from [10]; additional details are provided in the Appendix.

**Theorem 2.3.** *Let  $\varphi$  be a fixed encoder mapping each input  $X$  to a representation  $T = \varphi(X)$ , and let  $\Delta$  denote the generalization gap of a classifier built on top of  $Z$ . Then, for a dataset of size  $n$ , the generalization gap is bounded by:*

$$\Delta \leq \tilde{O}\left(\sqrt{\frac{I(X; T|Y)}{n}}\right).$$

In our setting, Notably, when  $\beta = 1$ , the criterion in Equation 1 can be reduced to:

$$\mathcal{J}_i = I(d_i; TK_i) - I(TK_i; A_t) = I(d_i; TK_i | A_t),$$

with the details provided in the Appendix. Therefore, suppose we have two candidate demonstrations  $d_i$  and  $d_j$ , and the selection criterion favors  $d_i$ , i.e.,  $I(d_i; TK_i | A_t) < I(d_j; TK_j | A_t)$ . According to the generalization bound from Theorem 2.3, which scales with  $I(d; TK | A_t)$ , we obtain:

$$\tilde{O}\left(\sqrt{\frac{I(d_i; TK_i | A_t)}{n}}\right) < \tilde{O}\left(\sqrt{\frac{I(d_j; TK_j | A_t)}{n}}\right).$$

Thus, selecting demonstrations based on our proposed criterion provably yields a tighter generalization bound. This result reinforces our central claim: our method enhances in-context learning performance by isolating transferable knowledge and reducing the influence of task-specific noise.

## 2.4 Transferable Knowledge Estimation and Demonstration Selection

At each timestep  $t$ , our goal is to select a demonstration  $d$  from a pool of candidates  $\mathcal{D}$  using the criterion in Equation 1. Since the mutual information terms are generally intractable to compute directly, we introduce practical strategies to approximate them.

First, we employ a pre-trained LLM as the *Knowledge Retriever*, inducing a stochastic channel  $d \mapsto TK_d$  with fixed capacity. This leads to a constant mutual information  $I(d; TK_d)$  across all  $d \in \mathcal{D}$ , simplifying the objective to maximizing  $I(TK_d; A_t)$ , which encourages selecting demonstrations whose transferable knowledge is most predictive of the next action.

Then, since  $A_t$  is the action to be predicted, it is not observable at the time of demonstration selection. To address this, we employ the same *Knowledge Retriever* to extract the anticipated transferable knowledge from the current agent context  $H_t$  denoted as  $TK_t$ , as a proxy of  $A_t$ . Then, to estimate the mutual information  $I(TK_d, TK_t)$ , we apply the InfoNCE lower bound [27], which leads to the following retrieval objective:

$$d^* \approx \arg \max_{d \in \mathcal{D}} \log \frac{\exp(\text{sim}(TK_d, TK_t))}{\sum_{d' \in \mathcal{D}} \exp(\text{sim}(TK_{d'}, TK_t))},$$

where  $\text{sim}(\cdot, \cdot)$  denotes a similarity function, computed using cosine similarity. These approximations effectively bridge the gap between our demo selection criteria formulation (Equation 1) and a practical, efficient implementation for dynamic demonstration selection.

### 3 Experiment

#### 3.1 Experimental Setting

**Tasks** Following prior work [57, 33, 63], we evaluate our method on two diverse and challenging domains, including reasoning tasks and sequential decision-making tasks. For reasoning tasks, we evaluate on HotpotQA[55], a multi-hop question answering dataset, whereas for sequential decision making tasks, we evaluate on Webshop [56], an interactive web-based shopping environment, and AlfWorld [35], a text-based embodied decision-making environment. Detailed descriptions of the settings for each task are elaborated in Sections 3.2–3.3.

**Baselines** We integrate our proposed method with several existing agentic frameworks, including ReAct [57], Reflexion [33], and LATS [63]. For each baseline, we adhere strictly to the original implementation details and hyperparameter settings to ensure fair and controlled comparisons. Specifically, for Reflexion, we use a trial count of 5, and for LATS, we adopt the LATS(ReAct) configuration as described in the original paper.

**LLM Agent** In line with previous work [33], all experiments are conducted using the *gpt-3.5-turbo* model. To ensure fair comparison, we reproduce some baseline results using *gpt-3.5-turbo* when the original work used models such as *text-davinci-002*, which are no longer supported. The reproduced results are consistent in magnitude with the original reports.

**Knowledge Retriever** To extract transferable knowledge, we employ the open-source language model *gemma-2-2b-it* [39] as the pre-trained encoder. Full details of the prompting strategy used to derive TK are provided in the Appendix.

**Demonstration Pool Construction** Our method requires a pool of candidate demonstrations from which to retrieve. For each task, we construct this pool by running a small subset of task instances using the respective baseline agent and collecting only the successful trajectories. During evaluation, we use the remaining, non-overlapping task instances to assess performance.

#### 3.2 Reasoning: HotpotQA

**Setup** HotPotQA [55] is a multi-hop question answering benchmark based on Wikipedia, designed to test an agent’s ability to retrieve and reason over multiple documents to answer complex questions. Following the setup from [57], we use a simplified Wikipedia environment that supports three interaction primitives: Search[entity], Lookup[string] and Finish[answer]. Performance is measured using the standard Exact Match (EM) score. We evaluate all methods on a 500-question subset. For other settings, such as the number of few-shot demonstrations, we adhere strictly to the exact settings specified in each baseline method.

**Result** Our experimental results are presented in Table 1. Across all baselines, integrating our method yields consistent and non-trivial improvements in Exact Match (EM) score. These gains highlight the effectiveness of our approach in enhancing agentic reasoning across different frameworks, demonstrating its generality and robustness. Notably, the improvements hold across agent architectures, suggesting our method offers benefits orthogonal to model-specific optimizations.

Table 1: Performance on HotpotQA. The number denotes the percentage of Exact Match (EM).

	EM ↑
ReAct	32.1
ReAct + DICE	41.4 ↑(+9.3)
Reflexion	51.6
Reflexion + DICE	58.9 ↑(+7.3)
LATS	63.3
LATS + DICE	71.4 ↑(+8.1)

#### 3.3 Sequential decision making: ALFWorld & Webshop

**AlfWorld** [35] is a suite of interactive, text-based environments adapted from the ALFRED [34] benchmark, where an agent completes multi-step household tasks via natural language commands. These tasks span six categories: Pick, Clean, Heat, Cool, Look, and Pick Two. The agent interacts with a simulated home by issuing actions like go to, take, or use, navigating complex environments with multiple rooms and objects. Each task may require over 50 steps, demanding effective planning, subgoal tracking, and commonsense reasoning. Following prior work, we evaluate our method on 134 unseen evaluation task instances using Success Rate (SR) as the performance metric.

Table 3: Results on AlfWorld with Success Rates (SR) across subcategories. Our method consistently improves performance across all subcategories and achieves a significant margin over the baseline.

	Pick	Clean	Heat	Cool	Look	Pick 2	All
ReAct	66.7	38.7	82.6	76.2	55.6	23.5	57.5
ReAct + DICE	79.2	51.6	87.0	76.2	66.7	47.1	67.9 $\uparrow(+10.4)$
Reflexion	83.3	58.1	91.3	90.5	72.2	52.9	74.6
Reflexion + DICE	87.5	71.0	95.7	90.5	83.3	64.7	82.1 $\uparrow(+7.5)$

**WebShop** [56] is a language-based environment simulating an online shopping platform, where agents are tasked with selecting a product that satisfies a user’s natural language query, such as “a black desk chair under \$100 with lumbar support”. Agents interact via search queries, button selections, and page navigation, simulating web browsing behavior. Performance is measured using two metrics: **success rate**, and the **average score**, which captures the average reward obtained across episodes. In line with previous work[57, 33], we use the test set of 500 shopping tasks for evaluation and maintain consistency with established action space definitions and prompting strategies.

**Result** The results are shown in Table 3 and Table 2 for AlfWorld and Webshop respectively. The result shows that integrating our method into existing agentic frameworks leads to consistent performance improvements across both benchmarks. On AlfWorld, we observe substantial gains in success rates across all subcategories, especially in more challenging tasks like “Clean” and “Pick 2”. Similarly, on Webshop, our approach improves both average score and success rate across all baselines. These results demonstrate the broad applicability and effectiveness of our method in enhancing agent performance in diverse and complex environments.

Table 2: Score and success rate (SR) on Webshop. Our method consistently improves performance across all baseline methods by a significant margin.

	Score $\uparrow$	SR $\uparrow$
ReAct	53.8	28.0
ReAct + DICE	58.2 $\uparrow(+4.4)$	35.0 $\uparrow(+7.0)$
Reflexion	64.2	35.0
Reflexion + DICE	67.4 $\uparrow(+3.2)$	38.2 $\uparrow(+3.2)$
LATS	75.9	38.0
LATS + DICE	77.1 $\uparrow(+1.2)$	39.5 $\uparrow(+1.5)$

### 3.4 Ablation Study & Other Comparisons

A key feature of our method is stepwise demonstration selection, i.e., dynamically choosing different demonstrations at each timestep as the agent progresses through a task. To evaluate the benefit of this design, we conduct an ablation study comparing it with a task-level variant, where a fixed set of demonstrations selected by DICE at timestep 0 is used throughout the task.

We also compare against established selection methods including KATE [15], which uses a  $k$ NN-based strategy to select relevant demonstrations, and EPR [30], which trains an additional module to facilitate selection. While both methods were originally proposed for one-step question answering, they lack adaptation for multi-step agentic settings. Thus, we treat them as task-level selectors in our experiments.

The results, summarized in Table 4, highlight the following: (1) Taskwise DICE outperforms existing methods, underscoring the importance of isolating transferable knowledge (TK). While EPR performs comparably, it requires additional training, whereas our method is entirely training-free.(2) Stepwise DICE consistently outperforms its taskwise variant, demonstrating the value of adapting demonstrations at each reasoning step. This confirms the effectiveness of our approach and shows that stepwise selection yields meaningful gains beyond what static, task-level selection can achieve.

Table 4: Comparison on Taskwise & Stepwise demo selection methods, evaluated by Exact Match(EM) on HotpotQA, SR for AlfWorld and Webshop.

	HotpotQA	AlfWorld	Webshop
ReAct	32.1	57.5	28.0
KATE	34.7	58.2	30.5
EPR	36.5	60.3	30.1
DICE (Taskwise)	36.3	61.2	31.3
DICE (Stepwise)	<b>41.4</b>	<b>67.9</b>	<b>35.0</b>

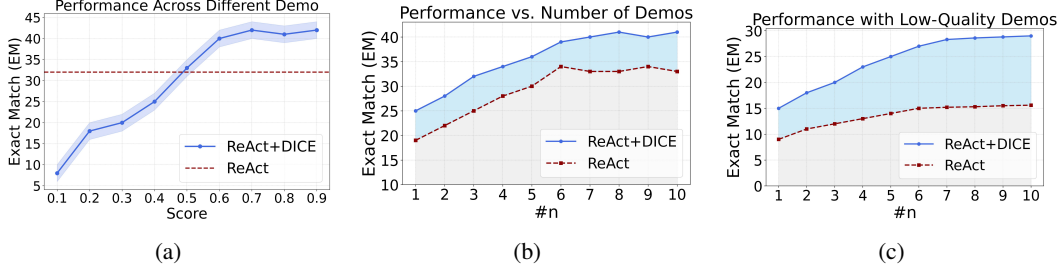


Figure 3: (a) Using demonstrations with higher average score in DICE results in better performance. (b) DICE outperforms standard ICL consistently with the same number of demonstrations, and matches its performance with significantly fewer examples. (c) DICE maintains an even stronger margin over standard ICL with only low-quality demos, demonstrating robustness and the ability to extract transferable knowledge from suboptimal inputs.

### 3.5 Further Analysis of DICE Performance

Beyond the main evaluations stated in previous sections, we conduct additional analyses to better understand the effectiveness and characteristics of our method. The results in this section are from on HotpotQA, evaluated by Exact Match (EM), results on other tasks are provided in the Appendix.

**How does better in-context example benefit improve performance?** Our method assigns a relevance score between 0 and 1 to each demo-target pair, reflecting how well a demonstration aligns with the target under our proposed selection criterion. To evaluate whether these scores correspond to actual utility, we group episodes based on the average score of their selected demonstrations and plot performance as a function of demonstration quality (Figure 3a). A horizontal red dashed line indicates the baseline performance under random demo selection (i.e., ReAct). As the figure shows, selecting higher-scoring demonstrations consistently leads to improved performance, clearly illustrating that our scoring metric is predictive of ICL effectiveness.

**How does our method reduce the number of required demonstrations?** By selecting higher-quality demonstrations, our method achieves comparable or better performance using fewer examples. As shown in Figure 3b, DICE consistently outperforms standard ICL across different numbers of demonstrations. Notably, with just 3 selected examples, DICE matches the performance of standard ICL that uses 6 randomly chosen ones. This highlights the efficiency of our approach in reducing reliance on large numbers of demonstrations without sacrificing effectiveness.

**What if there are no relevant demonstrations available?** In scenarios where no highly relevant demonstrations are available, it is important to assess whether our method still provides benefits. To simulate this, we constrain the demonstration pool to only include examples with a relevance score below 0.5 and compare the performance of DICE and standard ICL in Figure 3c. Interestingly, we observe that the performance gap in favor of DICE becomes even larger under these challenging conditions. This result suggests that our method remains effective even with suboptimal demonstrations—likely due to its principled focus on maximizing transferable knowledge, which enables better utilization of available examples despite their lower overall quality.

### 3.6 Case Study

To qualitatively illustrate the advantages of our approach, we present some representative examples from the experiments, as shown in Figure 4. In the first case, ReAct fails to identify the answer even though it appears directly in the observation, while ReAct + DICE correctly extracts the answer without unnecessary steps. This suggests that our method is better at leveraging useful information already found during the problem-solving process. In the second case, ReAct issues several overly specific and unproductive search queries, while ReAct + DICE quickly adjusts its strategy to a more effective search term and successfully retrieves the correct answer. This demonstrates our method’s ability to guide the agent toward more relevant queries when the initial attempt fails. These



<b>Question: What is the middle name of the Silicon Valley entrepreneur discussed in the book The Unwinding?</b>	
<b>ReAct:</b> Action 1: Search[The Unwinding]; Observation 1: The Unwinding: An Inner History .... Action 2: Search[Peter Thiel] Observation 2: Peter <b>Andreas</b> Thiel is ... Action 3: <b>Lookup[middle name]</b> Observation 3: No more results. <b>Ignored the name</b> .... Action 5: Finish[unknown]	<b>ReAct + DICE:</b> Action 1: Search[The Unwinding]; Observation 1: The Unwinding: An Inner History .... Action 2: Search[Peter Thiel] Observation 2: Peter <b>Andreas</b> Thiel is ... Action 3: <b>Finish[Andreas]</b> <b>Successfully identified the middle name</b>
<b>Question: Who is the mother of the Texas businessman that co-founded Ignite! Learning, Inc?</b>	
<b>ReAct:</b> Action 1: Search[Ignite! Learning, Inc.] Observation 1: Could not find ... Action 2: <b>Search[co-founders of Ignite! Learning, Inc.]</b> Observation 2: Could not find .... Action 3: <b>Search[notable figures associated with Ignite! Learning]</b> <b>Unproductive search</b> Observation 3: Could not find ... .... Action 8: <b>Finish[unknown]</b>	<b>ReAct + DICE:</b> Action 1: Search[Ignite! Learning, Inc.] Observation 1: Could not find ... <b>More effective</b> Action 2: <b>Search[Ignite! Learning]</b> <b>search strategy</b> Observation 2: Ignite! Learning, Inc. is a software company co-founded Neil Bush ... Action 3: Search[Neil Bush] Observation 3: Neil Mallon Bush is ... He is the fourth of six children of former President George H. W. Bush and Barbara Bush ... Action 4: <b>Finish[Barbara Bush]</b>

Figure 4: Case studies. **Top:** Without DICE, the agent overlooks the correct answer already present in Observation 2 and unnecessarily attempts a Lookup[middle name], resulting in failure. With DICE, the agent correctly extracts the middle name directly from the observation. **Bottom:** Without DICE, the agent issues a series of unproductive search queries, while DICE enables the agent to identify and search for the correct term by the second step, leading to a successful answer.

examples highlight how integrating DICE into the agent framework leads to more accurate and efficient reasoning in challenging scenarios, and improves reasoning and decision-making.

## 4 Related Work

**Agentic Framework** Agentic frameworks empower LLMs to solve long-horizon tasks by enabling observation reading, natural-language planning, and tool use [57, 31, 33, 63, 21, 60, 25]. ReAct [57] shows that interleaving reasoning and actions improves performance over purely reasoning- or action-based approaches. Toolformer [31] enables zero-shot API usage by letting the model decide when to invoke tools without extra supervision. Reflexion [33] introduces a self-critique loop to reduce repeated errors. LATS [63] unifies reasoning, acting, and planning via a tree search algorithm.

**In-Context Learning & Demo Selection** In-context learning (ICL) enables language models to solve new tasks by conditioning on a few input–output examples without weight updates [2]. Recent works enhance ICL via specialized pretraining [7, 14, 32] or intermediate training stages [23, 9, 46, 47, 3]. However, the effectiveness of ICL is highly sensitive to the choice and order of demonstrations, leading to research on calibration [24, 62], ordering [19], and rationale-based prompting [49]. Adaptive methods address this by selecting examples per query using compression objectives [51] or feedback-trained retrievers [43]. Theoretical studies also analyze the effectiveness of ICL [1, 4, 12].

Recent works have explored various demo selection strategies [5]. Unsupervised methods typically retrieve nearest neighbours based on input similarity [15, 38, 28], or use model output scores as selection criteria [26, 13, 50]. Mutual information [36], perplexity [6] and graphs [37] have also proven effective for prompt selection. Alternatively, some approaches train auxiliary models to guide selection [30, 58, 44, 40, 59, 45]. However, these require supervision and additional training, limiting their applicability in practice. Other approaches refine prompts iteratively or rank examples by difficulty or uncertainty [29, 41, 53], and [61] dynamically adjusts prompt length to avoid performance drops from excessive examples. Other techniques such as Demonstration Reformatting [11, 8, 16] and Demonstration Ordering [17, 18] are also explored. However, these works focus on non-agentic settings. While recent efforts target agentic tasks [20], they are domain-specific and lack theoretical grounding. In contrast, our DICE framework offers a general, theoretically grounded solution for dynamic demonstration selection in multi-step agentic tasks.

## 5 Conclusion

We introduced DICE, a dynamic and theoretically grounded in-context learning framework designed to enhance the performance and robustness of LLM-based agents in multi-step reasoning and tool-use tasks. By modeling the demonstration selection process through a causal lens, we identified how non-transferable knowledge in examples can introduce spurious dependencies and hurt generalization. Our proposed method dynamically selects the most relevant demonstrations at each reasoning step, maximizing transferable knowledge while operating entirely at inference time and requiring no additional training. Empirical results across diverse benchmarks and agentic frameworks validate the effectiveness of our approach, showing consistent gains over strong baselines. One limitation of our work is that we only explore a single instantiation of the framework; more expressive implementations—such as incorporating a trainable encoder to capture latent transferable knowledge—are left for future exploration. A broader impact of our approach is the potential to enable user-customized agents that learn from tailored examples, supporting more intuitive and adaptive human-AI interaction.

## References

- [1] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [3] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.
- [4] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, 2023.
- [5] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- [6] Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022.
- [7] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-training to learn in context. *arXiv preprint arXiv:2305.09137*, 2023.
- [8] Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2022.
- [9] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-impl: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.
- [10] Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In *International Conference on Machine Learning*, pages 16049–16096. PMLR, 2023.
- [11] Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*, 2022.
- [12] Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. The closeness of in-context learning and weight shifting for softmax regression. *Advances in Neural Information Processing Systems*, 37:62584–62616, 2024.
- [13] Xiaonan Li and Xipeng Qiu. Finding supporting examples for in-context learning. *CoRR*, 2023.
- [14] Yichuan Li, Xiyao Ma, Sixing Lu, Kyumin Lee, Xiaohu Liu, and Chenlei Guo. Mend: Meta demonstration distillation for efficient and effective in-context learning. *arXiv preprint arXiv:2403.06914*, 2024.

- [15] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.
- [16] Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*, 2023.
- [17] Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. Let’s learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024.
- [18] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.
- [19] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, 2022.
- [20] Michael Lutz, Arth Bohra, Manvel Saroyan, Artem Harutyunyan, and Giovanni Campagna. Wilbur: Adaptive in-context learning for robust and accurate web agents. *arXiv preprint arXiv:2404.05902*, 2024.
- [21] Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172*, 2023.
- [22] Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. *arXiv preprint arXiv:2305.14264*, 2023.
- [23] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.
- [24] Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, 2022.
- [25] Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, et al. Gui agents: A survey. *arXiv preprint arXiv:2412.13501*, 2024.
- [26] Tai Nguyen and Eric Wong. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*, 2023.
- [27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [28] Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*, 2023.
- [29] Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455, 2024.
- [30] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.
- [31] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.
- [32] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Gergely Szilvasy, Rich James, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, et al. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*, 2023.
- [33] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

- [34] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Motlaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.
- [35] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.
- [36] Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*, 2022.
- [37] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.
- [38] Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. Multilingual llms are better cross-lingual in-context learners with alignment. *arXiv preprint arXiv:2305.05940*, 2023.
- [39] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
- [40] Minh-Hao Van, Xintao Wu, et al. In-context learning demonstration selection via influence analysis. *arXiv preprint arXiv:2402.11750*, 2024.
- [41] Duc Anh Vu, Nguyen Tran Cong Duy, Xiaobao Wu, Hoang Minh Nhat, Du Mingzhe, Nguyen Thanh Thong, and Anh Tuan Luu. Curriculum demonstration selection for in-context learning. *arXiv preprint arXiv:2411.18126*, 2024.
- [42] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.
- [43] Liang Wang, Nan Yang, and Furu Wei. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, 2024.
- [44] Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 1:15, 2023.
- [45] Xubin Wang, Jianfei Wu, Yichen Yuan, Mingzhe Li, Deyu Cai, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. *arXiv preprint arXiv:2412.03966*, 2024.
- [46] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.
- [47] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [48] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [50] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*, 2022.

- [51] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436, 2023.
- [52] Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in neural information processing systems*, 30, 2017.
- [53] Shangqing Xu and Chao Zhang. Misconfidence-based demonstration selection for llm in-context learning. *arXiv preprint arXiv:2401.06301*, 2024.
- [54] Chen Yang, Chenyang Zhao, Quanquan Gu, and Dongruo Zhou. Cops: Empowering llm agents with provable cross-task experience sharing. *arXiv preprint arXiv:2410.16670*, 2024.
- [55] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.
- [56] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.
- [57] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- [58] Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR, 2023.
- [59] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, 2022.
- [60] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.
- [61] Fei Zhao, Taotian Pang, Zhen Wu, Zheng Ma, Shujian Huang, and Xinyu Dai. Dynamic demonstrations controller for in-context learning. *arXiv preprint arXiv:2310.00385*, 2023.
- [62] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR, 2021.
- [63] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.