

# PIVOT-ICL: ADAPTIVE EXEMPLAR SELECTION FOR IN-CONTEXT LEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

In-context learning (ICL) enables LLMs to better complete complex tasks, but the key to success relies on selecting the right exemplars. Prior work often selects exemplars based on similarity to the test input, but this can fail when no close match exists—especially for challenging or out-of-distribution examples. In such cases, generic exemplars that broadly represent the task may be more helpful. This raises two key questions: (1) Which exemplars best represent the overall task distribution? (2) Which test examples are too far from available exemplars? We propose Pivot-ICL, a method that adaptively selects either dynamic, input-specific exemplars or static, generic exemplars. We model this decision using a bipartite graph between test examples and candidate exemplars, applying well-established graph mining algorithms such as Hyperlink-Induced Topic Search (HITS) to enable weighted, iterative, and bidirectional scoring. This graph-based voting identifies both representative exemplars and which test inputs require them. Experiments show Pivot-ICL consistently improves performance across tasks, achieving a +8.8% relative gain over the best baseline. Further analysis attributes the improvement to effective adaptive treatment of both in-distribution and out-of-distribution examples.

## 1 INTRODUCTION

Large language models (LLMs) show great performance with in-context learning (ICL), on various complex tasks, e.g., commonsense reasoning (Zhang et al., 2023), math (Agarwal et al., 2024), and planning (Zhao et al., 2025). Compared to fine-tuning task-specific models, ICL reduces the need for extra training and potentially makes LLMs more versatile. However, one key to successfully deploying ICL is the selection of the exemplars (Nie et al., 2022; Zhang et al., 2022).

Previous work explores various ways of selecting exemplars to conduct successful ICL, from manual selection (Zhao et al., 2021; Lu et al., 2022b; Chang & Jia, 2022), to automatically decide specific exemplars for each test example (Rubin et al., 2022; Zhang et al., 2022; Ye et al., 2023a). Besides such *dynamic* selection on similar questions, recently, researchers have explored the effectiveness of automatically selecting a generic set of *static* exemplars for each task (Li & Qiu, 2023; Purohit et al., 2024). Despite the significant performance on various tasks, however, given a specific set of exemplar candidates, there is no guarantee that these candidates can cover all kinds of test examples in the same task. As a result, some test examples can benefit less when we use certain exemplar selection methods with a unified treatment.

Specifically, some test examples may have similar exemplars, where an effective dynamic selection method can potentially benefit the model reasoning by providing a curated set. Meanwhile, some other test examples can diverge from the collected exemplar candidates, where a static set of exemplars eliciting the general task format and understanding can be better than providing potentially misleading dynamic exemplars. Above cases depict our intuition - besides the uni-directional evaluation of exemplars with each test example, the opposite is also important for the strategic deployment of ICL, which constitutes bidirectional relations, so that we can (1) reflect the exemplar importance based on how they are used; and (2) decide which test examples are considered distant.

To capture such bilateral interactions between exemplar and test example sets, we propose to model them from a bipartite graph view, as shown in Figure 1. The edges between exemplar or test example nodes will be weighed via the relevance among them. In this way, with graph mining algorithms

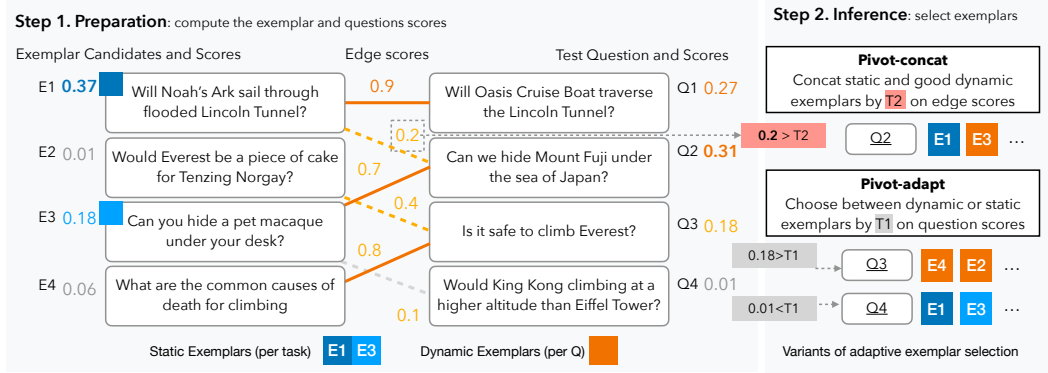


Figure 1: An example of our Pivot-ICL pipeline on questions adapted from SQA (Geva et al., 2021). With the bipartite graph edges as similarities across exemplars and questions, we compute their corresponding exemplar and question scores with HITS. With Pivot-concat, we concatenate the static exemplars and the top dynamic exemplars. With Pivot-adapt, based on thresholding the question (hub) scores, we choose either dynamic (closest) or static (generic) exemplars for the questions.

such as Hyperlink-Induced Topic Search (HITS, Kleinberg, 1999), when scoring the test examples, beyond simply averaging similarity from all exemplars, their importance and characteristics in the exemplar communities can also be taken into consideration in a weighted, bidirectional, and iterative importance computation across the two sets. On the one hand, we score the test examples that can reflect their relations towards a specific set of exemplars. On the other hand, the importance of the exemplars are also based on how frequently they support the test cases. From the graph view, we introduce Pivot-ICL, which offers adaptive treatment to different test examples. For questions with high scores, we utilize their closely relevant exemplars in a dynamic exemplar selection manner. For low score questions that are potentially distant from the exemplars, we use a task-level generic and static set of high-score exemplars to show models how to approach this task in general. As shown in Figure 1, for test examples in different situations, variants of Pivot-ICL strategically conduct different treatments. In Pivot-ICL, we aim to build zero-shot signals to decide which method to use for each specific test example.

We conduct extensive experiments on four complex tasks covering planning (Bohnet et al., 2024), mathematical reasoning (Mathematical Association of America, 2024), commonsense reasoning (Geva et al., 2021), and PhD-level scientific question answering (Rein et al., 2024). We show that Pivot-ICL achieves robust and consistent performance improvement (on average +8.8 percent, relatively) over baselines and can show better overall performance than solely using static/dynamic exemplar selection methods for all the queries.

We also validate the consistent gain across various closed- and open-sourced backbones. Analyses further show that the gain comes from this bilateral graph automatically recognizing which examples are more out of distribution and need to benefit from generic exemplars.

In summary, our main contributions are:

- We propose to model the bilateral interactions between exemplars and test examples to allow adaptive treatment on different questions in ICL exemplar selection.
- We design the bilateral weighted graph and Pivot-ICL to allow score assignment for both the exemplars and test examples, which serves as an effective zero-shot adaptive treatment and shows good performance.
- Further analysis shows how the performance gain of Pivot-ICL comes from the different treatment of in-distribution and out-of-distribution test examples.

## 2 RELATED WORK

**LLM In-Context Learning.** Prompting and in-context learning (ICL) have been considered as the prominent ways to interact with LLMs, which achieve significant performance on various tasks (Wei et al., 2022; Brown et al., 2020). Recent work also discovers the extendability of ICL on tasks requiring complex reasoning, such as math (Agarwal et al., 2024), planning (Zhao et al., 2025), and agent interactions (Lutz et al., 2024), where one key to the success of ICL is exemplar selection.

Previous work studies how perturbation (Zhang et al., 2022) and learned models (Rubin et al., 2022; Ye et al., 2023a) helps assign good exemplars for each example. Ye et al. (2023b); Zhang et al. (2023) further highlights the importance of diversity among the selected exemplars. Besides the dynamic exemplar selection that assigns question-specific exemplars, LENS (Li & Qiu, 2023) and EXPLORA (Purohit et al., 2024) further explore the potential to use a static set of exemplars across the whole task. More recently, ConE (Peng et al., 2024) studies the impact of model conditional entropy, which leads to good ICL performance if open-weight models are available.

In our work, we propose to leverage the bilateral relations among exemplars and test examples to allow zero-shot adaptive treatment on different questions based on their scores from the perspective of the exemplar set. We utilize the embedding-based similarity scores that do not rely on computing the entropy of the question/answer given exemplars.

**Example Selection in General.** Selecting the crucial examples has been a long-lasting topic in the machine learning community. Support Vector Machine (SVM) (Cortes & Vapnik, 1995) mines a few support vectors to conduct classification. Corset Selection (Guo et al., 2022; Mirzasoleiman et al., 2020) aims to select a set of training examples to improve the end tasks, in a data-efficient manner (Adadi, 2021). With similar motivations, a set of few-shot adaptation methods is proposed to improve LLM performance on various textual reasoning tasks (Houlsby et al., 2019; Gao et al., 2020; Tam et al., 2021; Li & Liang, 2021; Logan IV et al., 2022).

Our work is based on a similar motivation for selecting important examples in the ICL pipeline. Compared to these previous works, we do not monitor the gradients of the large language models and focus on using semantic similarity to model the relations among exemplars and test examples in a zero-shot manner.

### 3 PIVOT-ICL

#### 3.1 THE ICL TASK AND THE STATUS-QUO

In this paper, we investigate how to improve the in-context learning (ICL) performance on various complex reasoning tasks with large language models (LLMs). Specifically, given sets of exemplar candidates ( $c \in C$ ) and test examples ( $q \in Q$ ). Our goal is to find a list of good exemplars (i.e., shots,  $\{c_1, c_2, \dots\}$ ) to be put in the context with prompts. The LLMs will then solve the test examples with the demonstrative context. We note that, as shown in (Zhang et al., 2023), to allow successful ICL on complex reasoning tasks, the exemplar decoration may further include the rationales of task solving, besides questions and corresponding answers that are typically used in general ICL (Min et al., 2022). In this sense, the exemplars can provide context beyond the simple format of the tasks, e.g., how to approach certain questions.

Following the notation of (Purohit et al., 2024), as introduced in Section 1, previous work explores different kinds of exemplar selection methods: (1) *dynamic* selection in a question-specific manner, e.g., k-nearest neighbors (KNN) selection (Rubin et al., 2022) - for each test example  $q_i$ , there are specific  $\{c_1^i, c_2^i, \dots\}$  used; (2) *static* selection for the whole task, e.g., Auto-CoT (Zhang et al., 2023) selects diverse exemplars from different clusters - for the whole task, there are specific  $\{c_1, c_2, \dots\}$ .

#### 3.2 CORNER STONE: WEIGHTED BIPARTITE GRAPH

In realistic scenarios, there is often no perfect set of exemplar candidates that can cover all the categories of distribution of certain tasks. In response, we propose to capture the bilateral interactions between the exemplar and test examples from the point of view of the graph.

Our method is motivated by the mutual-reinforcement nature of the classic Hyperlink-Induced Topic Search (HITS, Kleinberg, 1999) algorithm. Originally, applying HITS to a website graph helps identify *authorities* (websites that contain valuable, trustworthy information on a topic) and *hubs* (websites that serve as curators of the information sources). The corresponding *authority* and *hub* scores are computed iteratively, i.e., previously identified *authorities* have higher weights on deciding next round *hubs*, vice versa. In our ICL context, exemplars are akin to *authorities* that potentially contain valuable information for the tasks. test examples can be *hubs* linked to the exemplars.

We formalize the graph construction of building a bipartite graph  $G = (C \cup Q, E, W)$ , where  $C$  represents the exemplars,  $Q$  represents test examples, and each edge  $e_{ij} \in E$  carry a weight  $w(q, c)$  computes the normalized similarity between  $q, c$ .

Extending HITS, we can compute the *authority* and *hub* scores for exemplars and test examples, respectively. The scores capture two key dynamics: (1) exemplars serving as authoritative sources of problem-solving patterns, and (2) test examples functioning as hubs that aggregate knowledge from multiple exemplars. The iterative computation follows:

$$\text{authority}(c) = \sum w(q, c) \times \text{hub}(q) \quad (1)$$

$$\text{hub}(q) = \sum w(q, c) \times \text{authority}(c) \quad (2)$$

This process models the adaptive in-context learning paradigm we propose: exemplars demonstrating broadly applicable patterns gain higher authority scores, while test examples capable of utilizing high-quality exemplars achieve higher hub scores. Convergence identifies the most influential exemplars and test examples most likely to benefit from the current combinations, optimizing selection and weighting for in-context learning.

With the framework above, to construct the node and edges, we utilize the direct way to construct the node of exemplars and test examples: the inputs of them to extract the representation as the nodes. Besides the surface-level semantic relevance among inputs, as shown in (Zhao et al., 2025), strong representation models (Lee et al., 2024; Muennighoff et al., 2025a; Weller et al., 2025; Cai et al., 2025) can potentially contain information about how to solve the task.

We link the nodes with weighted edges computed by embedding-based textual similarity (Reimers & Gurevych, 2019). Specifically, in our experiments, we use the cosine similarity of embeddings from Gecko (Lee et al., 2024), a lightweight and general-purpose strong representation model. For each query-exemplar pair, the edge weight is written as  $w(q, c)$ . To allow efficiency and reduce noise, we only keep the top-100 weighted edges for each test example node. Since we build a bipartite graph across exemplars and test examples, the relations among exemplars are modeled implicitly with test examples as proxies. We compare alternative design choices of node construction and edge building in Section 4.5.

### 3.3 EXEMPLAR SELECTION PER TEST EXAMPLE

With the weighted bipartite graph, we can achieve both dynamic and static selection per test example:

**Selecting Dynamic Sets.** From the graph view, selecting the most similar exemplars for each test example is the same as selecting the nodes linked with the highest weighted edges. Our current implementation is the same as similarity-based KNN selection (Rubin et al., 2022). We compare various edge weight assignments in Section 4.3.

**Selecting Static Sets.** Previous approaches to selecting static exemplar sets often rely on clustering in embedding space, such as KMeans (Zhang et al., 2023), which can be sensitive to outliers and may overlook nuanced task structures. Under the graph view, we instead treat static exemplar selection as a process of identifying structurally important nodes among exemplars, using classical graph mining techniques (Newman, 2005; Cook & Holder, 2006; Chakrabarti & Faloutsos, 2006).

As an interesting byproduct, the graph we built in Section 3.2 allows us to model exemplar-test interactions. This enables us to apply node scoring functions—such as degree centrality (Shaw, 1954), authority scores (Kleinberg, 1999), and PageRank (Page et al., 1999)—to rank exemplar nodes by their global influence within the graph. Formally, given a scoring function  $\phi$  and graph  $G$ , each exemplar  $c_i$  receives a score  $s_i^C = \phi(G, c_i)$ . The top-ranked exemplars form the static set.

### 3.4 PIVOT BETWEEN DYNAMIC AND STATIC

Once both dynamic and static exemplar sets are constructed, we introduce two strategies to adaptively or jointly leverage them for each test example: **Pivot-concat** and **Pivot-adapt**. We first introduce Pivot-concat as a novel dynamic exemplar selection method that augments the original ICL pipeline with static exemplars, with extra consideration on interactions between exemplars and test examples.

Then, we introduce Pivot-adapt as a new ICL paradigm to automatically switch between dynamic and static exemplars.

**Pivot-concat** maintains the original ICL pipeline and fuses the most relevant dynamic exemplars with the most representative static exemplars. This method allows static examples to complement the dynamic ones, particularly when dynamic matches are sparse or shallow.

Pivot-concat uses edge-weight thresholds for filtering. For a given test example  $q$ , let  $C^d = \{c_1^d, c_2^d, \dots, c_k^d\}$  be its dynamic exemplars sorted by edge weight, and  $C^t = \{c_1^t, c_2^t, \dots, c_k^t\}$  the static exemplars ranked by node score  $s_i^C$ . We filter the dynamic set using a threshold  $t_o$ , keeping only those with  $w(q, c_m^d) \geq t_o$ , where  $t_o = \mu_q + 2 * \sigma_q$  of the edge weights for  $q$  to keep the statistically significant dynamic exemplars.

The final exemplar set for  $q$  is the concatenation  $C^d + C^t$ . This method avoids explicit scoring of test examples and instead augments strong local matches with task-level exemplars for added robustness. Compared to the adaptive version, Pivot-concat is essentially a dynamic way to select the exemplars without scoring the exemplars explicitly, e.g., with *hub* scores.

**Pivot-adapt** dynamically chooses between the dynamic and static sets on a per-example basis, based on how well a test example is connected to exemplar nodes in the graph. Well-connected examples receive tailored, input-specific dynamic exemplars; less connected or outlier examples are paired with static exemplars that broadly capture task-level patterns.

Specifically, for each test example  $q_i$ , we compute a graph-based score  $s_i^Q = \phi(G, q_i)$  using a chosen graph-based scoring function  $\phi$ , which we compare against a threshold  $t_\nabla$ . If  $s_i^Q \geq t_\nabla$ , we assign the dynamic exemplar set to  $q_i$ ; otherwise, we use the static set. We set  $t_\nabla = \alpha / (|C| + |Q|)$ , where  $\alpha$  is a hyperparameter<sup>1</sup>, and  $|C|$  and  $|Q|$  normalize the score with the sizes of the exemplar and test sets, respectively. This method provides a principled way to decide whether local or global context is more beneficial for a given input.

We also note that the test example scores here can work with any pairs of *dynamic* and *static* exemplar selection methods, which is not limited by the specific method we used to compute the scores. We further validate our design of the thresholding mechanism for both variants in Appendix A.5.

## 4 EXPERIMENTS AND ANALYSIS

### 4.1 CHALLENGING TASKS AND SETUP

Motivated by Agarwal et al. (2024), we focus on evaluating Pivot-ICL on complex tasks that require extensive reasoning and planning for problem solving. The tasks and ICL setups are as follows:

- **AIME24**. AIME24 (Mathematical Association of America, 2024) has 30 problems that were used in the 2024 American Invitational Mathematics Examination, which tests the mathematical problem-solving capability on topics ranging from arithmetic, algebra, counting, geometry, and etc. We adapt the 877 math reasoning questions in **s1K-1.1** (Muennighoff et al., 2025b) to serve as the candidate exemplars, where each question contains a reasoning trace and a solution.
- **PDDL**. PDDL (Aeronautiques et al., 1998) is a standardized programming language used in various communities to represent planning problems, with problems defined with an initial state and an environment. The tasks require models to generate a sequence of actions that will satisfy the final goals. Following the PDDL benchmarks (Bohnet et al., 2024), we test the LLM performance with **Blocksworld** and use the 28,000 exemplars (3-7 blocks). The test examples are the in-distribution (PDDL-ID, 3-7 blocks) and out-of-distribution (PDDL-OD, 8-20 blocks) splits, with 300 questions each. ID split is used in our main experiment.
- **StrategyQA**. SQA (Geva et al., 2021) is a challenging commonsense reasoning task with questions that typically require models to know specific factual knowledge and utilize it to answer multiple multi-hop sub-questions beforehand. We follow (Purohit et al., 2024) to split the original training set to 1,800 exemplars and 490 test examples. We test a challenging setting where we do not provide models with the gold facts or sub-questions for the test examples.

<sup>1</sup>empirically, we set  $\alpha$  to 2000 across tasks, which can be optimized with a development set.

Table 1: Performance comparison with 10 exemplars. *dynamic* and *static* denote the corresponding exemplar selection methods. Standard and CoT use no exemplar. Average denotes the macro-average among the four tasks. **Bold** denotes the best-performing entry per column.

| Mode                | Methods                       | PDDL        | AIME24      | SQA         | GPQA        | Average     |
|---------------------|-------------------------------|-------------|-------------|-------------|-------------|-------------|
| <i>Dynamic only</i> | BM25 (Trotman et al., 2014)   | 58.7        | 10.0        | 82.7        | 62.1        | 53.4        |
|                     | SimCSE (Gao et al., 2021)     | 57.3        | 13.3        | 82.0        | 61.1        | 53.4        |
|                     | Gecko (Lee et al., 2024)      | 61.3        | 20.0        | 80.8        | 60.6        | 55.7        |
|                     | MMR (Ye et al., 2023b)        | 57.6        | 10.0        | 79.8        | 60.6        | 52.0        |
| <i>Static only</i>  | Standard                      | 2.3         | 13.3        | 71.0        | 23.7        | 27.6        |
|                     | CoT (Kojima et al., 2022)     | 0.0         | 10.0        | 64.2        | 38.4        | 28.2        |
|                     | Random                        | 43.0        | 10.0        | 79.1        | 57.1        | 47.3        |
|                     | Auto-CoT (Zhang et al., 2023) | 46.3        | 13.3        | 83.2        | 53.5        | 49.1        |
|                     | Degree Centrality             | 49.0        | 16.7        | 81.4        | <b>66.7</b> | 53.5        |
|                     | Authority (Kleinberg, 1999)   | 47.3        | 20.0        | 81.0        | 62.6        | 52.7        |
| <i>Pivot (ours)</i> | PageRank (Page et al., 1999)  | 52.0        | 16.7        | 80.6        | 64.1        | 53.4        |
|                     | Pivot-concat                  | 60.3        | 20.0        | 82.2        | 62.1        | 56.2        |
|                     | Pivot-adapt                   | <b>69.3</b> | <b>23.4</b> | <b>83.9</b> | 65.6        | <b>60.6</b> |

- **GPQA.** GPQA (Rein et al., 2024) is a challenging dataset with 448 multi-choice questions that require expert knowledge in the domains of biology, physics, and chemistry. We use 198 PhD-level science questions in **GPQA Diamond** as the test examples, and the rest 250 questions and the exemplar candidates.

**Setup.** Unless otherwise specified, we use Gemini 1.5 Pro (Gemini Team et al., 2024) as the default backbone to generate answers. We evaluate with a temperature of 1 and a max number of 4,096 output tokens. We use the standard metrics of the tasks, i.e., exact match (EM) for AIME24 and GPQA, cover-EM (Press et al., 2023) for SQA, and planning accuracy (Bohnet et al., 2024) for PDDL.

## 4.2 BASELINES

We set various *dynamic* and *static* exemplar selection methods as our baselines. We focus on comparing with similar approaches that rely solely on the representation of the exemplar candidates and test examples, and are without the need for computing losses from the LLMs.

For *dynamic* methods, we consider the KNN-alike methods (Rubin et al., 2022) with different ways to compute the similarity between each pair of exemplars and test examples:

- BM25 (Trotman et al., 2014), with weighted token-level match;
- SimCSE (Gao et al., 2021), with cosine similarity of unsupervised sentence embeddings;
- Gecko (Lee et al., 2024), with cosine similarity of distilled embeddings from LLMs.

We also include MMR (Ye et al., 2023b) that considers the diversity among selected exemplars. We implement MMR with Gecko similarity.

For *static* methods, we first compare with random exemplar selection and clustering-based Auto-CoT (Zhang et al., 2023). With the graph view of the relations among exemplars and test examples we introduced, we are further allowed to use classic graph mining methods to choose a set of static exemplars based on the ranks, e.g., the degrees of exemplar nodes (Degree), Authority scores (Kleinberg, 1999), and PageRank (Page et al., 1999). We also compare with non-ICL methods: directly promoting the model in a QA format (Standard) or with zero-shot COT (Kojima et al., 2022).

## 4.3 MAIN RESULT: EFFECTIVENESS OF PIVOT-ICL

We present the performance of our Pivot-ICL and various baselines in Table 1. Pivot-adaptive here denotes the combination of *Gecko* and *Authority* for exemplar selection with *hubs* scores. Pivot-concat. here denotes the fused exemplars from *Gecko* and *Authority* through statistical thresholds  $t_\circ$ , instead of thresholding on *hub* scores with  $t_\nabla$ .

**Graphs help effective static exemplar selection.** Table 1 shows that, through our graph view of the relations among exemplar candidates and test examples, simple and classic graph mining methods, i.e., Degree Centrality, Authority, and PageRank, can all achieve better overall performance compared to Auto-CoT, which demonstrates the importance of considering the bilateral relations.

**Variants of Pivot-ICL achieve a consistently good performance.** Table 1 presents that the proposed Pivot-adapt achieves the best performance on average through automatically selecting whether to use the dynamic or static sets of exemplars based on the graph-based scores assigned to the test examples. On tasks such as PDDL, AIME24, and SQA, we can observe that our query-level adaptive treatment achieves better overall performance than each individual method, which demonstrates the benefit of Pivot-ICL. As for Pivot-concat, we can observe that, this new dynamic exemplar selection method shows good overall performance that surpasses Gecko on SQA and GPQA. The only gap comes from PDDL, where an abundant set of exemplars is designed to be in the same distribution as the test examples.

On the other hand, comparing these two variants, Pivot-adapt still offers significantly better performance than Pivot-concat with the additional step of scoring the test examples with the graph view, instead of relying solely on the distribution of the similarity between test examples and the top-ranked exemplars. Such observations further highlight the importance of proposing adaptive treatment in the Pivot-ICL paradigm beyond the existing dynamic and static exemplar selection methods.

### Pivot-ICL generalizes to different backbones.

We further examine the generalization of Pivot-ICL with different backbone language models. To allow meaningful performance comparison for different-sized models, we conduct the backbone ablation on SQA and GPQA, which are Boolean True or False QA and multiple-choice QA, respectively. For static and dynamic baselines, we use the best-performing ones in the main experiments on these two tasks, i.e., Gecko and Degree.

We extend our experiments with other backbones including Gemini 2.0 Flash, Llama 3.3 (Llama-3.3-70B-Instruct, Grattafiori et al., 2024)<sup>2</sup>, and Qwen 2.5 (Qwen-2.5-7B-Instruct, Qwen et al., 2025).

From Table 2, we can observe that, across different tasks and backbone LLMs, our proposed Pivot-ICL consistency outperforms using dynamic exemplars per question or a set of static exemplars per task, which demonstrates the generalizability of our method.

**Pivot-ICL is comparable with loss-based methods in a zero-shot manner.** Our method focuses on pivoting between exemplar sets in a zero-shot manner purely based on the data representations. Here, we further compare with other state-of-the-art static exemplar selection methods that require more training or inference time computations: (1) LENS (Li & Qiu, 2023) that utilizes the LLM output probabilities; and (2) EXPLORA (Purohit et al., 2024) that proposes a sampling-based bandit algorithm that learns a scoring function from the exemplar subset losses.

We follow the original settings of (Purohit et al., 2024) to test on GSM8K (Cobbe et al., 2021), TabMWP (Lu et al., 2022a), and SQA\*<sup>3</sup> with GPT-4o-mini as the backbone and 5 exemplars. Pivot-adapt here denotes the combination of *Gecko* and

Table 2: Performance on SQA and GPQA with different backbone large language models. *dynamic*, *static*, and *pivot* denote *Gecko*, *Authority*, and *Pivot-adapt* in the main experiments, respectively. The best-performing entry on each column is marked in **bold**.

| Backbone         | Method  | SQA         | GPQA        |
|------------------|---------|-------------|-------------|
| Gemini 2.0 Flash | dynamic | <b>82.2</b> | 62.1        |
|                  | static  | 79.8        | 60.1        |
|                  | pivot   | <b>82.2</b> | <b>64.1</b> |
| Llama 3.3 70B    | dynamic | 77.8        | 41.9        |
|                  | static  | 76.5        | 42.4        |
|                  | pivot   | <b>78.6</b> | <b>44.4</b> |
| Qwen 2.5 7B      | dynamic | 70.6        | 30.8        |
|                  | static  | 69.6        | 31.3        |
|                  | pivot   | <b>71.2</b> | <b>35.4</b> |

Table 3: Performance comparison with 5 exemplars following the original settings. Results of LENS and EXPLORA are extracted from the original paper (Purohit et al., 2024).

| Method      | GSM8K | TabMWP | SQA* |
|-------------|-------|--------|------|
| LENS        | 76.2  | 86.3   | 92.9 |
| EXPLORA     | 93.6  | 90.1   | 95.1 |
| Pivot-adapt | 93.1  | 94.0   | 92.0 |

<sup>2</sup>All authors with industry affiliation only advised on the project and did not participate in the use of Llama models nor conducted any experimentation.

<sup>3</sup>\* denotes an easier setting of SQA compared to our main experiments, where the oracle facts and sub-questions are given for the test examples

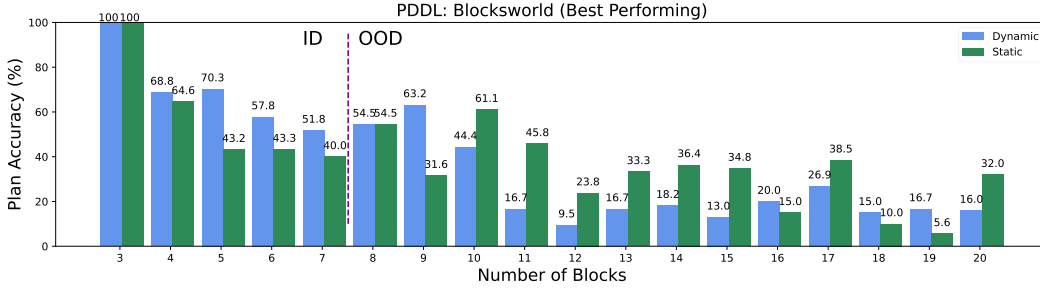


Figure 2: ICL performance with different numbers of blocks. *Best Performing* denotes the best micro-average performance for PDDL and PDDL-ood tasks – Gecko for *Dynamic* and Auto-CoT for *Static*, respectively. Since all the exemplar candidates are with 3-7 blocks, we denote test examples with 3-7 blocks as in-distribution (ID) and 8-20 blocks as out-of-distribution (OOD).

*Authority* for exemplar selection with *hub* scores. Table 3 shows that Pivot-ICL achieves comparable performance with these loss-based exemplar selection methods. However, loss-based methods are expensive to adapt to each task, and can potentially be limited by insufficient exemplar candidates as the development set. In contrast, Pivot-ICL provides efficiency and easy adaptation. We further discuss the difference in Appendix A.6.

#### 4.4 ID VS. OOD: AN IN-DEPTH ANALYSIS ON WHY THE METHOD WORKS

Our method builds upon the intuition that, given a fixed set of exemplars, the test examples shall be treated differently since they may or may not be covered by the exemplars. In this section, we further quantitatively verify this intuition with a controlled experiment. As described in Section 4.1, with the original splits of the PDDL task, we can clearly define an in-distribution set and an out-of-distribution set based on the numbers of blocks in the problem, where all the demonstrative exemplars have 3-7 blocks. Following the original setting (Bohnet et al., 2024), we set the split line to 7<sup>4</sup> and compare the dynamic and static exemplar selection methods that has the best overall performance on PDDL plus PDDL-OOD, i.e., Gecko and Auto-CoT.

From Figure 2, we can observe that, for the in-distribution questions, the dynamic method (blue bars) shows consistently better performance than the static method (green bars) across different numbers of blocks. However, when the questions are out-of-distribution, where it is less likely to find similar exemplars, we can observe that the static exemplars perform better than the dynamic ones. The above observation motivates how our Pivot-ICL work: for tasks without a clear boundary of in-distribution or out-of-distribution, we utilize the interactions of the exemplars and test examples to build a graph to automatically decide between which set to rely on.

#### 4.5 ABLATION STUDIES ON GRAPH CONSTRUCTION

In this section, we further explore the extendability of Pivot-ICL with the ablation of different design choices for each component. The results are presented in Table 4.

**Alternative node construction.** Beyond direct node construction with inputs (*exemplar*, *test*), we further consider variants with more contextual information. Specifically, following Zhang et al. (2023), we include additional *rationales* to the exemplars, so as to better demonstrate the task completion principles. In this case (we denote *ex+r*), each exemplar node is represented as { *exemplar input*, *rationales*, and *exemplar output* }. Moreover, to facilitate better mappings between the exemplars and the test cases, we further adapt generative resampling (Zhao et al., 2025) — We randomly select some exemplars, use them to prompt the LLM to generate some initial answers and rationales, and then concatenate LLM-generated rationales to test case nodes as well. We denote this construction as *t+r*. Table 4 shows different pairings between node constructions.

From the table, we see that both (*ex*, *t*) and (*ex+r*, *t+r*) show good performances, highlighting the importance of uniform construction methods for both exemplars and test cases. Although the exemplar prompts can provide additional rationales, the heterogeneous approach (*ex+r*, *t*) does not always improve performance.

<sup>4</sup>This is not necessarily a hard split but an indicator of the distributional differences. Problems with smaller block sizes are often straightforward, while more blocks can lead to the need for paralleled working threads.

**Alternative edge construction.** For edge building, besides the default version with a *bipartite* graph (bi) we can also use the edges among exemplars (*exampler*), or include edges both between exemplars and test cases as well as among exemplars (bi+ex). From Table 4, we can observe that our default method with the bipartite graph achieves similar performance with the exemplar-only variant (ex), but our bipartite graph would rely on way fewer edges, improving efficiency.

The good performance from the exemplar-only variant also validates the extendability of our method to the no-test-example scenarios, where we can still extract a good set of static exemplars with the candidates alone, despite the additional cost. Similar to our findings for node construction, we can also observe that the hybrid method bi+ex degrades the performance, which indicates the potential noise introduced by the distributional differences.

**Alternative weighing method.** In Section 3.2, given a specific set of exemplar candidates and test examples, there is a fixed bilateral weighted graph. However, with generative resampling mentioned above, every time we attempt to generate a hypothetical rationale/answer for a test example  $t$ , we can reconstruct a new version of  $t+r$ , replacing the previous attempt — which can form an iterative loop between generative resampling and exemplar selection, with the node scores and edge weights changing every iteration. We further test out this alternative dynamic weighing, by doing one iteration after the default (ex+r, t+r).

From Table 4, as expected, we can identify that utilizing the model attempts can improve the performance, which is a good characteristic of generative resampling. However, iteration will cost additional calls of LLMs. We leave the exploration on the iterative methods or Self-Refine-alike methods (Madaan et al., 2023) with Pivot-ICL for future research.

## 5 CONCLUSION AND DISCUSSION

Finding a good context for the model to conduct efficient and effective problem-solving on complex tasks is crucial. This paper introduces a new perspective to view the exemplar selection process of in-context learning (ICL). Beyond previously proposed per-question and per-task exemplar selection methods, we propose to utilize the bilateral relations among exemplar candidates and test examples to allow adaptive treatment over the questions. Upon building the weighted bipartite graph, we design Pivot-ICL, a new paradigm that automatically decides whether to use question-specific exemplars or generic task-specific exemplars when the question is not likely covered, given the current exemplar candidates. Extensive experiments on four challenging tasks show the great performance of Pivot-ICL across backbone LLMs and settings. A closer look at the performance difference between in-distribution and out-of-distribution questions sheds light on the importance of the adaptive treatment strategy when there is less guarantee of good exemplars in real deployment.

We design Pivot-ICL based on a bipartite graph. In reality, sometimes, there is no observed full set of test examples. In these cases, we can use exemplars to perform k-fold validation to find the significant exemplars and thresholds for distant test examples. In the future, we plan to explore (1) self-evolving exemplar selection methods, e.g., extending the iterative node construction; (2) generative exemplar selection, e.g., when Pivot-ICL detects that a question is not sufficiently covered by current candidates, we can generate specific exemplars on the fly or based on the candidates, extending (Chen et al., 2023). We will open source our code at [url redacted for review].

Table 4: Ablation study on SQA and GPQA with different design choices. Node and Edge denote the ablation study for node construction and edge building, respectively. *-pg* denotes the use of exemplar prompt and generated outputs for test examples. Details of the methods are in Section 3

| Type      | Method                 | SQA  | GPQA |
|-----------|------------------------|------|------|
| Node      | Gecko (ex, t)          | 80.8 | 60.6 |
|           | Gecko (ex+r, t)        | 81.0 | 57.1 |
|           | Gecko (ex+r, t+r)      | 81.0 | 59.1 |
| Edge      | Authority (bipartite)  | 81.0 | 62.6 |
|           | Authority (exemplar)   | 82.4 | 62.6 |
|           | Authority (bi+ex)      | 80.2 | 63.1 |
| Iteration | Gecko (ex+r, t+r)      | 81.0 | 59.1 |
|           | Gecko (ex+r, t+r)-iter | 81.0 | 61.1 |

## REFERENCES

- Amina Adadi. A survey on data-efficient algorithms in big data era. *Journal of Big Data*, 8(1):24, 2021.
- Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins Sri, Anthony Barrett, Dave Christianson, et al. Pddl the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.
- Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie C.Y. Chan, Biao Zhang, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL <https://openreview.net/forum?id=goi7DFHlqS>.
- Bernd Bohnet, Azade Nova, Aaron T Parisi, Kevin Swersky, Katayoon Goshvadi, Hanjun Dai, Dale Schuurmans, Noah Fiedel, and Hanie Sedghi. Exploring and benchmarking the planning capabilities of large language models. *arXiv preprint arXiv:2406.13094*, 2024.
- Tom B Brown, Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1, 2020.
- Fengyu Cai, Tong Chen, Xinran Zhao, Sihao Chen, Hongming Zhang, Sherry Tongshuang Wu, Iryna Gurevych, and Heinz Koepl. Revela: Dense retriever learning via language modeling, 2025. URL <https://arxiv.org/abs/2506.16552>.
- Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR)*, 38(1):2–es, 2006.
- Ting-Yun Chang and Robin Jia. Data curation alone can stabilize in-context learning. *arXiv preprint arXiv:2212.10378*, 2022.
- Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. Self-ICL: Zero-shot in-context learning with self-generated demonstrations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15651–15662, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.968. URL <https://aclanthology.org/2023.emnlp-main.968/>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Diane J Cook and Lawrence B Holder. *Mining graph data*. John Wiley & Sons, 2006.
- Ryan A Cook, John P Lalor, and Ahmed Abbasi. No simple answer to data complexity: An examination of instance-level complexity metrics for classification tasks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2553–2573, 2025.
- Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.
- Jiale Fu, Yaqing Wang, Simeng Han, Jiaming Fan, and Xu Yang. Graphic: A graph-based in-context example retrieval model for multi-step reasoning, 2025. URL <https://arxiv.org/abs/2410.02203>.
- Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- Gemini Team et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*, 2024. URL <https://arxiv.org/abs/2403.05530>.

Mor Geva, Daniel Khoshabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. URL <https://api.semanticscholar.org/CorpusID:230799347>.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelier van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shao-liang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank

- 594 Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Sweeney,  
595 Gil Halpern, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hakan Inan,  
596 Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph,  
597 Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog,  
598 Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James  
599 Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny  
600 Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings,  
601 Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai  
602 Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik  
603 Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle  
604 Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng  
605 Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish  
606 Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim  
607 Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle  
608 Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang,  
609 Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam,  
610 Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier,  
611 Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia  
612 Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro  
613 Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani,  
614 Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy,  
615 Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin  
616 Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Sibi, Sai Jayesh Bondu,  
617 Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh  
618 Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay,  
619 Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang,  
620 Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie  
621 Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta,  
622 Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman,  
623 Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehele, Thomas Robinson, Tianhe Li, Tianjun  
624 Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria  
625 Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru,  
626 Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz,  
627 Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv  
628 Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi,  
629 Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait,  
630 Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The  
631 llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- 632 Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection  
633 in deep learning. In *International Conference on Database and Expert Systems Applications*,  
634 pp. 181–195. Springer, 2022.
- 635 William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs,  
636 2018. URL <https://arxiv.org/abs/1706.02216>.
- 637 Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David  
638 Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti  
639 Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del  
640 Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren  
641 Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming  
642 with numpy. *Nature*, 585:357–362, 2020. doi: 10.1038/S41586-020-2649-2. URL <https://doi.org/10.1038/s41586-020-2649-2>.
- 643 Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe,  
644 Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning  
645 for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th  
646 International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning  
647 Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL <https://proceedings.mlr.press/v97/houlsby19a.html>.

- Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy S Liang, and Jure Leskovec. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems*, 36:16302–16317, 2023.
- John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03): 90–95, 2007.
- Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pp. 668–677. Society for Industrial and Applied Mathematics, 1999.
- Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftexhar Naim. Gecko: Versatile text embeddings distilled from large language models, 2024. URL <https://arxiv.org/abs/2403.20327>.
- Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9, 2024.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL <https://aclanthology.org/2021.acl-long.353/>.
- Xiaonan Li and Xipeng Qiu. Finding support examples for in-context learning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6219–6235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.411. URL <https://aclanthology.org/2023.findings-emnlp.411/>.
- Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting down on prompts and parameters: Simple few-shot learning with language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2824–2835, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.222. URL <https://aclanthology.org/2022.findings-acl.222/>.
- Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022a.
- 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 Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556. URL <https://aclanthology.org/2022.acl-long.556/>.

- Michael Lutz, Arth Bohra, Manvel Saroyan, Artem Harutyunyan, and Giovanni Campagna. Wilbur: Adaptive in-context learning for robust and accurate web agents, 2024.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=S37hOerQLB>.
- Mathematical Association of America. American invitational mathematics examination (AIME). Mathematical competition, 2024. URL [https://artofproblemsolving.com/wiki/index.php/AIME\\_Problems\\_and\\_Solutions/](https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions/).
- Sewon Min, Xinxin Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL <https://aclanthology.org/2022.emnlp-main.759/>.
- Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.
- Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning, 2025a. URL <https://arxiv.org/abs/2402.09906>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025b. URL <https://arxiv.org/abs/2501.19393>.
- Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54, 2005.
- Feng Nie, Meixi Chen, Zhirui Zhang, and Xu Cheng. Improving few-shot performance of language models via nearest neighbor calibration, 2022. URL <https://arxiv.org/abs/2212.02216>.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL <http://ilpubs.stanford.edu:8090/422/>. Previous number = SIDL-WP-1999-0120.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>.
- Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9090–9101, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.492. URL <https://aclanthology.org/2024.acl-long.492/>.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL <https://aclanthology.org/2023.findings-emnlp.378/>.

- Kiran Purohit, Venkatesh V, Raghuram Devalla, Krishna Mohan Yerragorla, Sourangshu Bhattacharya, and Avishek Anand. EXPLORA: Efficient exemplar subset selection for complex reasoning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5367–5388, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.307. URL <https://aclanthology.org/2024.emnlp-main.307/>.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410/>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=Ti67584b98>.
- Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL <https://aclanthology.org/2022.naacl-main.191>.
- Marvin E Shaw. Some effects of unequal distribution of information upon group performance in various communication nets. *The journal of abnormal and social psychology*, 49(4p1):547, 1954.
- Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*, 2021.
- Together AI Team. Together ai, 2024. URL <https://www.together.ai/>.
- Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium, ADCS '14*, pp. 58–65, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330008. doi: 10.1145/2682862.2682863. URL <https://doi.org/10.1145/2682862.2682863>.
- 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.
- Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-trained retrievers can be prompted like language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=odvSjn416y>.
- Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. *The Fortieth International Conference on Machine Learning*, 2023a.
- Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. Complementary explanations for effective in-context learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL*

2023, pp. 4469–4484, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.273. URL <https://aclanthology.org/2023.findings-acl.273/>.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active Example Selection for In-Context Learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9134–9148, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.622. URL <https://aclanthology.org/2022.emnlp-main.622>.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

Xinran Zhao, Hanie Sedghi, Bernd Bohnet, Dale Schuurmans, and Azade Nova. Improving large language model planning with action sequence similarity. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=tpGkEgxMJT>.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/zhao21c.html>.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

## A APPENDIX

### A.1 LIMITATIONS

**Advanced Graph Mining Techniques.** In this work, we propose to study the bilateral relations among exemplars and candidates. For node construction, we treat the whole exemplar or test example as an atomic unit. However, motivated by (Huang et al., 2023; Fu et al., 2025), each exemplar itself can be viewed as a subgraph as well. Considering the hierarchical relations among the overall graph and the subgraphs can be an interesting direction. On the other hand, our work focuses on proposing the bilateral weighted graph to conduct Pivot-ICL, so that we start with the classic node scoring methods like PageRank to avoid blurred contribution. Advanced graph mining methods, e.g., GraphSage (Hamilton et al., 2018), can be explored in the future to unlock advanced performance.

**Limited Exemplar Sets.** Our current ICL settings for AIME24 and GPQA are with limited sizes of the exemplar sets, i.e., 877 and 250 questions, respective. We conduct this pilot study to show the robustness of our proposed methods with limited exemplars. We anticipate potential improved performance on these challenging tasks with extended exemplars, e.g., through adding problems such as NumiaMATH (Li et al., 2024) or AGIEval (Zhong et al., 2023). Besides the sizes, another dimension to build an improved exemplar set is to add external knowledge that is associated with the questions, e.g., further provide summarization of the provided relevant web pages for GPQA.

### A.2 IMPLEMENTATION DETAILS

For Gemini and GPT models, we use the official API service. For Llama 3.3 and other open-sourced models, we use the instruct-turbo version provided by Together AI (Together AI Team, 2024) or a locally served model on a machine with 8 Nvidia A6000 (40G) GPUs with CUDA 12 installed with inference structure built upon vLLM (Kwon et al., 2023). If applicable, we set the max output token to be 4,096, temperature to be 1.0, top p to be 0.7, and top k to be 50. For the graph construction, we used networkx 3.4.2.

### A.3 THE USE OF LARGE LANGUAGE MODELS (LLMs).

In this work, LLMs are used for correcting grammatical errors in writing and coding. We do not use LLMs to write papers or construct the logic of the whole code base.

### A.4 DATASET EXAMPLES

We present the examples of the data we used as follows. We omit some details to save space. For full details, please refer to the original work.

#### PDDL (Blocksworld)

```
Input: (define (problem BW-rand-4) (:domain blocksworld-4ops) (:objects b3 b2 b1 b4) (:init (on b4 b1)
(clear b4) (clear b2) (on b2 b3) (handempty) (ontable b3) (ontable b1) )
Goal:(:goal (and (on b2 b3) (on b1 b2) (on b4 b1) )))
Plan: (unstack b4 b1) (put-down b4) (pick-up b1) (stack b1 b2) (pick-up b4) (stack b4 b1)
```

| Standard Deviation | 1    | 1.5  | 2 (current) | 2.5  |
|--------------------|------|------|-------------|------|
| GPQA               | 53.0 | 62.1 | 62.1        | 59.6 |

Table 5: Performance results for different standard deviation thresholds on GPQA for Pivot-concat

| Method         | AIME 24 | SQA  | GPQA |
|----------------|---------|------|------|
| Mean Threshold | 20      | 79.4 | 63.1 |
| Ours           | 23.4    | 83.9 | 65.6 |

Table 6: Comparison of mean threshold vs. our method across different tasks for Pivot-concat

**GPQA**

Question: Methylcyclopentadiene (which exists as a fluxional mixture of isomers) was allowed to react with methyl isoamyl ketone and a catalytic amount of pyrrolidine. A bright yellow, cross-conjugated polyalkenyl hydrocarbon product formed (as a mixture of isomers), with water as a side product. These products are derivatives of fulvene. This product was then allowed to react with ethyl acrylate in a 1:1 ratio. Upon completion of the reaction, the bright yellow color had disappeared. How many chemically distinct isomers make up the final product (not counting stereoisomers)?

Explanation: Methylcyclopentadiene exists as an interconverting mixture of 3 isomers ... Cyclopentadienes will react with ketones in the presence of amine base to dehydrate and form fulvenes. The CH<sub>2</sub> group of the cyclopentadiene will lose 2 hydrogens over the course of the reaction to form the C=C bond, so the Methylcyclopentadiene will not react in the 5-methyl-1,3-cyclopentadiene form. Choices: (A) 2; (B) 16; (C) 8; (D) 4

**SQA**

Question: Are more people today related to Genghis Khan than Julius Caesar?

Facts: Julius Caesar had three children; Genghis Khan had sixteen children; Modern geneticists have determined that out of every 200 men today has DNA that can be traced to Genghis Khan.

Decompositions: How many kids did Julius Caesar have? How many kids did Genghis Khan have? Is #2 greater than #1?

Answer: True

**AIME24**

Question: Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of  $s$  kilometers per hour, the walk takes her 4 hours, including  $t$  minutes spent in the coffee shop. When she walks  $s + 2$  kilometers per hour, the walk takes her 2 hours and 24 minutes, including  $t$  minutes spent in the coffee shop. Suppose Aya walks at  $s + \frac{1}{2}$  kilometers per hour. Find the number of minutes the walk takes her, including the  $t$  minutes spent in the coffee shop.

Solution: Let  $W$  be the time Aya spends walking, in hours. We are given that when Aya walks at  $s$  kilometers per hour, the total time is 4 hours, so  $W + \frac{t}{60} = 4$ . Since the distance is 9 kilometers, we have  $\frac{9}{s} = W$ . Substituting this into the first equation gives  $\frac{9}{s} + \frac{t}{60} = 4$ . The total time is 3 hours plus  $t = 24$  minutes, which is 3 hours and 24 minutes.  $3 \times 60 + 24 = 180 + 24 = 204$  minutes. Final Answer: The final answer is 204

Answer: 204

**A.5 DIFFERENT THRESHOLD FOR PIVOT-CONCAT AND PIVOT-ADAPT**

In this section, we conduct further ablations on the thresholding mechanism of Pivot-concat and Pivot-adapt as follows:

For Pivot-concat, currently, we use mean + 2 standard deviations ( $\sigma$ ). Additionally, we test different standard deviation choices on GPQA with our current experiment setting in Table 5. The results show

that there is a sweet spot for the threshold, which can be optimized for each task. We use  $2\sigma$  across tasks. Practitioners can choose either better performance (Pivot-Aaapt) or easier adaptation of the pipeline (Pivot-concat).

For Pivot-adapt, the current threshold is set with  $\alpha$  and normalization from the data sizes, which already shows good performance across various datasets. In practice, we can use k-fold cross-validation over the exemplar sets to find an optimal  $\alpha$  for Pivot-adapt to achieve improved performance. We further validate the thresholding design with mean thresholds.

The consistent performance gain in Table 6 suggests the importance of setting a threshold that is relevant to the dataset statistics, as in our current practice.

#### A.6 COMPUTATION OVERHEAD

In this section, we further discuss the computational complexity of our methods. The extra computational overhead for Pivot-ICL, compared to current embedding-based exemplar selection methods, is computing the graph-based scores and visiting the scores ( $O(1)$  on average with a hash table) to decide which strategy of exemplar selection to use. For HITS, the time complexity is  $O(k \times (|V| + |E|))$ , where  $k$  is the number of iterations (typically 10-50),  $|V|$  is the number of nodes, and  $|E|$  is the number of edges (in our design,  $|E| \leq 100|V|$ ). The extraction of the graph-based scores can be done on CPU and in parallel, which creates little overhead compared to the inference of LLMs. In our experiments, it typically takes less than 10 minutes, depending on the size of the graph.

For loss-based exemplar selection, for example, EXPLORA (Purohit et al., 2024) typically requires 1000+ LLM calls and 2–5 hours as the overhead.

#### A.7 PROMPT EXAMPLES

In our work, we use generic prompts without heavy engineering. The keywords in  $\{\}$  are the same as what we presented in the examples in Section A.4. The prompts are as follows.

For PDDL exemplars, we use (“Please solve the problem: {Input}; You plan as plain text without formatting: {Plan}; done.”).

For AIME24 exemplars, we use (“Please solve the problem: {question}; Answer: {Solution}”).

For SQA exemplars, we use (“Question: {question}; Facts: {facts}; Decomposition: {decomposition}”).

For GPQA exemplars, we use (“Question: {question}; Choices: {choices}; Explanation: {explanation}; The correct answer is {answer}”).

For test examples, in our main experiment, we remove everything after the inputs or questions.

#### A.8 EXTENDED FUTURE WORK DISCUSSION

**Edge weights.** The edge weights are limited to cosine similarity explored in our current scope. From the cosine similarity perspective, one future work is to use prompt to adjust the embeddings, as discussed in Weller et al. (2025). Non-representation-based scores, e.g., influence scores (Koh & Liang, 2017), losses (Purohit et al., 2024), action sequence similarity for planning in (Zhao et al., 2025) can also be included. Beyond the relations among nodes, the instance-level hardness (Cook et al., 2025) can also be leveraged to adjust the weights based on the complexity of specific nodes. We leave the investigation on this line for future research in the community.

#### A.9 ETHICAL STATEMENTS

We foresee no ethical concerns or potential risks in our work. All of the retrieval models and datasets are open-sourced, as shown in Section 4. The LLMs we applied in the experiments are also publicly available. Given our context, the outputs of LLMs are unlikely to contain harmful and dangerous information. The experiments in our paper are mainly on English.

| Artifacts/Packages | Citation                                    | Link                                                                                                                                | License                |
|--------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| AIME24             | (Mathematical Association of America, 2024) | <a href="https://huggingface.co/datasets/HuggingFaceH4/aime_2024">https://huggingface.co/datasets/HuggingFaceH4/aime_2024</a>       | cc-by-sa-4.0           |
| PDDL               | (Bohnet et al., 2024)                       | <a href="https://arxiv.org/pdf/2406.13094">https://arxiv.org/pdf/2406.13094</a>                                                     | cc-by-sa-4.0           |
| SQA                | (Geva et al., 2021)                         | <a href="https://huggingface.co/datasets/voidful/StrategyQA">https://huggingface.co/datasets/voidful/StrategyQA</a>                 | MIT License            |
| GPQA               | (Rein et al., 2024)                         | <a href="https://github.com/idavidrein/gpqa">https://github.com/idavidrein/gpqa</a>                                                 | MIT License            |
| PyTorch            | (Paszke et al., 2019)                       | <a href="https://pytorch.org/">https://pytorch.org/</a>                                                                             | BSD-3 License          |
| numpy              | (Harris et al., 2020)                       | <a href="https://numpy.org/">https://numpy.org/</a>                                                                                 | BSD License            |
| matplotlib         | (Hunter, 2007)                              | <a href="https://matplotlib.org/">https://matplotlib.org/</a>                                                                       | BSD compatible License |
| vllm               | (Kwon et al., 2023)                         | <a href="https://github.com/vllm-project/vllm">https://github.com/vllm-project/vllm</a>                                             | Apache License 2.0     |
| Llama-3            | (Grattafiori et al., 2024)                  | <a href="https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct">https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct</a> | LICENSE                |
| Qwen 2.5           | (Qwen et al., 2025)                         | <a href="https://huggingface.co/Qwen/Qwen2.5-7B">https://huggingface.co/Qwen/Qwen2.5-7B</a>                                         | apache-2.0             |

Table 7: Details of datasets, major packages, and existing models we use. The datasets we reconstructed or revised and the code/software we provide are under the MIT License.

#### A.10 LICENSES OF SCIENTIFIC ARTIFACTS

We conclude the licenses of the scientific artifacts we used in Table 7. All of our usage for scientific discovery follows the original purpose of the artifacts.