# DPS: Diverse Prototype Selection for Adaptive In-Context Learning

Xuanbo Fan[1], Kaiyuan Li[2], Hao Sun[1], Boci Peng[1],
Zhenrong Cheng[1], and Yan Zhang[1] (✉)

[1] School of Intelligence Science and Technology, Peking University, China
{xuanbo.fan,sunhao,bcpeng,chengzhenrong}@stu.pku.edu.cn, zhyzhy001@pku.edu.cn
[2] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
likaiyuan2001@gmail.com

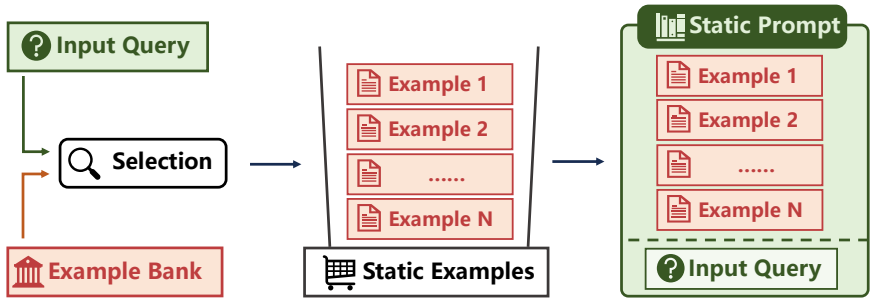
**Abstract.** Large language models exhibit remarkable proficiency across a wide array of tasks by leveraging in-context learning, wherein they learn from a limited number of examples. However, the efficacy of ICL is highly sensitive to the choice of demonstrations provided. Existing approaches primarily focus on the selection of individual examples, often neglecting the broader context of the entire example bank. In this paper, we introduce a novel framework aimed at augmenting the example bank through **D**iverse **P**rototype **S**election (**DPS**). *DPS* decomposes the ICL process into two distinct stages: *Prototype Selection* and *Prompt Synthesis*. In the first stage, DPS identifies a set of prototype functions that closely approximate the underlying data distribution. In the second stage, these prototype functions dynamically generate query-specific demonstrations, thus guiding the LLM more effectively in its task. Empirical evaluations conducted across thirteen reasoning benchmarks demonstrate that **DPS** significantly enhances ICL performance, providing substantial improvements when integrated with downstream LLMs.

**Keywords:** In-context Learning · Few-shot Learning.

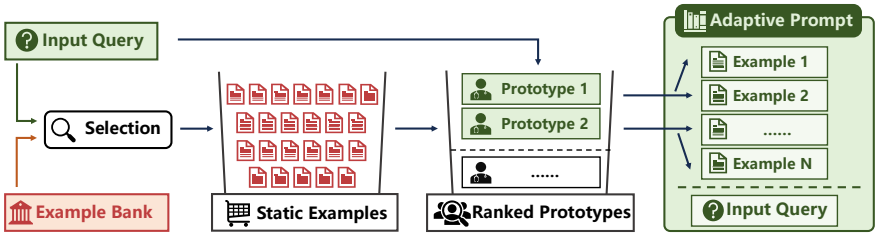
## 1 Introduction

Large language models (LLMs) [14] have achieved remarkable success across a broad range of natural language processing tasks [23], owing to their exceptional emergent capabilities. One of the most prominent emergent abilities is in-context learning (ICL), which utilizes a small number of input-output examples to enhance model predictions [4]. ICL has proven to be highly effective in unlocking the advanced potential of LLMs and has become a widely adopted strategy for tackling complex tasks.

However, due to the constraints imposed by the context window [10], only a limited number of examples can be incorporated into the prompt. Prior research [16, 13] has also demonstrated that ICL is highly sensitive to the selection and ordering of chosen examples [8], with even minor changes leading to significant performance fluctuations. Consequently, a key area of research has been the selection of high-performing demonstrations from the example bank.




(a) Inference pipeline of Static In-context Learning.




(b) Inference pipeline of Adaptive In-context Learning.

Fig. 1: Comparison of Static and Adaptive In-context Learning. Unlike Static ICL, which concatenates a small number of fixed inference examples into the prompt, Adaptive ICL first evaluates a large set of similar examples to determine the optimal prototype, then use them to generate diverse and adaptive examples.

A prominent line of research in example selection is the development of heuristic evaluation metrics to assess candidate examples. Some works focus on selecting examples that exhibit higher similarity to the input query [16], while other approaches aim to balance relevance and diversity [37]. Despite their effectiveness, these methods are inherently based on subjective judgments, which limits their robustness across different task scenarios. To address these limitations, iterative frameworks such as SE2 [15] and ConE [22] refine the selection process by incorporating feedback from downstream LLMs, enabling more context-aware adjustments and improving the adaptability of example selection across varying task demands.

Despite some success, current methods primarily focus on selecting **intact** examples from the example bank. However, the quality of the example bank is often overlooked. Since it is typically human-annotated, constructing the example bank is both time-consuming and costly, resulting in a smaller and less diverse set of examples that may fail to cover all potential scenarios. This limited selection space restricts the model's ability to access a sufficiently varied set of examples. Moreover, the reasoning paths in these human-labeled examples are typically single-faceted. Even when multiple valid approaches exist for a given problem, the solutions in the example bank usually follow a single predefined

path. This overreliance on fixed frameworks hinders the model's flexibility, preventing it from adapting effectively to new tasks. Therefore, rather than solely relying on pre-existing samples from the example bank, it is crucial to improve the **adaptiveness** and **diversity** of demonstrations for input queries by dynamically generating contextually relevant examples that align with different reasoning perspectives, as shown in Fig. 1. This enables the model to explore multiple approaches to problem-solving instead of rigidly following a single predefined path, enhancing its ability to tackle diverse and unfamiliar tasks.

Nevertheless, generating adaptive demonstrations for in-context learning is a non-trivial task due to several inherent challenges. **First**, human-annotated examples inherently limit the model's generation diversity. On the other hand, without human-annotated examples, LLM-generated results may not align with task-specific preferences, compromising the effectiveness of reasoning paths. Although increasing temperature and performing multiple sampling rounds can introduce diversity to some extent, the model's reasoning ability remains constrained by its intrinsic capabilities. **Second**, the absence of ground-truth labels during inference complicates the assessment of different demonstrations. Without clear evaluation criteria, it becomes challenging to determine which demonstrations are truly effective, making it difficult to identify the most suitable examples for in-context learning.

To address these challenges, as shown in Fig. 2, we propose a novel framework that enhances in-context learning through **D**iverse **P**rototype **S**election (**DPS**). **To ensure the diversity of the generated examples**,*DPS* first collects a set of reasoning patterns that approach queries from diverse perspectives, referred to as *prototype functions* (Section 4.1). It then utilizes the traditional example bank to evaluate and select the most suitable prototype functions upon receiving a user query. This is achieved through the construction of a *prototype bank* (Section 4.2), where *DPS* identifies similar problems and assesses different prototype functions to generate diverse reasoning paths. **To ensure the quality of the selected examples**, *DPS* employs two advanced reranking techniques: *frequency-based reranking* and *decay-based reranking*. These techniques refine the model's output by effectively leveraging consensus across the selected prototype functions, enhancing both accuracy and reasoning diversity (see Section 4.4).

Experimental results on thirteen datasets across three tasks demonstrate *DPS*'s effectiveness in significantly improving the performance of downstream LLMs. For instance, in mathematical reasoning tasks, an average improvement of 11.9% was achieved. To summarize, our contributions are as follows:

- We introduce *DPS*, an effective framework that leverages the prototype bank to generate demonstrations with diverse perspectives, adaptively tailored to the input query.
- To better leverage the inferences among prototype functions, we propose two advanced techniques: frequency-based reranking and decay-based reranking. These techniques further refine the selection of high-quality demonstrations.
- Extensive experimental results across thirteen reasoning datasets demonstrate the effectiveness of *DPS* compared to existing methods.

## 2 Related Work

While large language models have demonstrated impressive zero-shot performance across a variety of tasks, including complex reasoning and agent-based tasks [31, 25], recent studies show that in-context learning can further harness their potential and enhance their performance [4]. In addition to improving effectiveness, ICL can provide structural guidance that helps mitigate prompt bias during model inference [35]. Due to the constraints imposed by the context window [17], only a limited number of examples can be incorporated into the prompt. Previous research [16, 13] has also demonstrated that ICL is highly sensitive to the selection and ordering of examples [8].

Some studies focus on selecting examples with greater similarity to the input query [16], while others strive to balance both relevance and diversity [37]. Although these methods have shown promise, they are often based on subjective criteria, limiting their generalization across different tasks. To overcome these challenges, iterative frameworks such as SE2 [15] and ConE [22] introduce feedback loops from downstream LLMs, allowing for more contextually sensitive adjustments and improving the selection process for various task requirements.

In contrast to these approaches, our method removes the constraint of relying solely on an example bank for selecting in-context examples. Instead, we propose generating task-specific demonstrations using existing models, offering a fresh perspective on how ICL capabilities can be leveraged more effectively for downstream tasks.

## 3 Preliminary

Consider a downstream task $T$ that involves an example bank $B$, which consists of a set of input-output pairs $\{(x_n, y_n)_{n=1}^N\}$, and a pre-trained LLM with fixed parameters $\theta$. For a given input query $x_t$, the LLM generates an output $y_t$ by sampling from the following distribution:

$$y_t \sim \mathrm{LLM}_{\theta,\tau}\left[\mathrm{Demo}(x_t, B) \oplus x_t\right] \tag{1}$$

Here, $\tau$ represents the sampling temperature, which controls the randomness of the model's predictions. The function $\mathrm{Demo}(x_t, B)$ selects a sequence of examples from $B$ based on $x_t$ to generate demonstrations, and $\oplus$ denotes the concatenation of these examples with the input query $x_t$.

In subsequent sections, we will omit the fixed parameters $\theta$ and assume $\tau = 0$, which corresponds to greedy decoding (i.e., choosing the most likely output at each step). The goal of in-context learning is to design the $\mathrm{Demo}(x_t, B)$ algorithm to optimize performance for the task $T$.

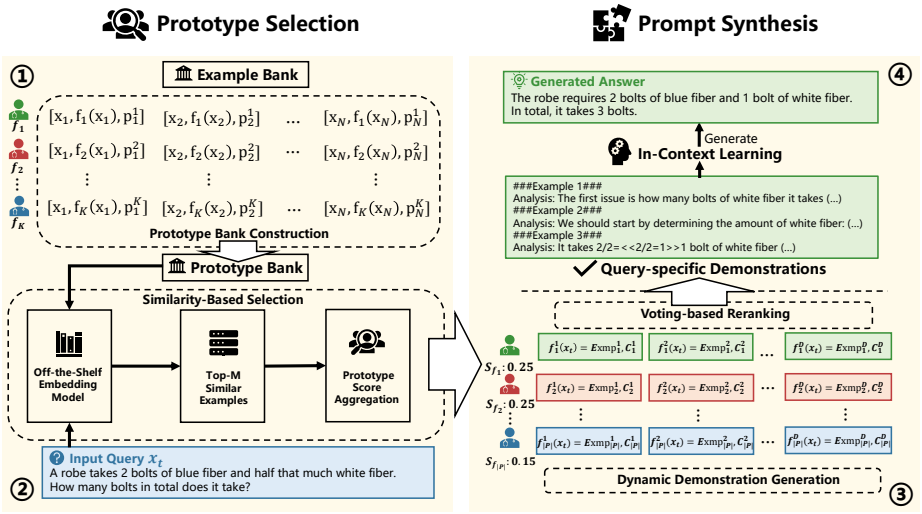



Fig. 2: The *DPS* pipeline consists of two main stages: (1) Prototype Selection: selecting a diverse set of prototype functions optimized to provide various reasoning perspectives, and (2) Prompt Synthesis: leveraging these prototype functions to generate adaptive demonstrations tailored to the input query.

# 4 Methodology

## 4.1 Overview

As shown in Fig. 2, we provide a comprehensive overview of our framework. Unlike conventional in-context learning methods that rely on static example selection, *DPS* dynamically selects and synthesizes demonstrations, optimizing reasoning from multiple perspectives. This framework consists of two key stages: *Prototype Selection* and *Prompt Synthesis*.

In the Prototype Selection stage, we first employ a set of specialized models with diverse prompting methods as *prototype functions*. The example bank is then reorganized into a *prototype bank*, which serves as the foundation for evaluating different prototype functions. Upon receiving a user query, *DPS* retrieves similar examples from the prototype bank, evaluates the effectiveness of different prototype functions based on their responses, and selects the most suitable ones for the query.

During the Prompt Synthesis stage, *DPS* prompts the selected prototype functions to generate adaptive demonstrations tailored to the input query. Additionally, two voting-based reranking strategies are employed to refine the selection process, ensuring that only the most effective demonstrations are used as contextual examples. The following sections will provide a detailed breakdown of each component of the *DPS* framework.

### 4.2 Prototype Bank Construction

Existing example selection methods work by selecting examples from an example bank based on similarity and then directly concatenating them into the prompt. These examples are considered perfectly accurate as they are manually annotated. Thus, the example bank is formally defined as:

$$B = \{(x_n, F(x_n), 1)_{n=1}^{N}\}, \tag{2}$$

where $F(x_n)$ represents the ground truth, and the label 1 confirms its correctness.

However, the high cost of manual annotation limits the example bank to a single analytical perspective, reducing its informational diversity. To mitigate this limitation, we expand the example bank by incorporating reasoning strategies from multiple perspectives. Specifically, we introduce several prototype functions as alternatives to ground truth and track their accuracy $p$ in answering these questions. This reconstructed example bank, referred to as the *Prototype Bank*, is formally defined as:

$$B' = \{(x_n, f_k(x_n, prompt_k), p_n^k)_{k=1}^{K}{}_{n=1}^{N}\}, \tag{3}$$

where $f_k$ refers to the $k$-th prototype function, $prompt_k$ refers to the $k$-th prompt method and $p_n^k$ indicates the accuracy of $f_k$ in answering $x_n$.

### 4.3 Prototype Selection

Given an input query $x_t$, we begin by embedding it using a pre-trained model, then calculate its similarity with other examples:

$$s_n^t = \frac{\text{Emb}(x_n) \cdot \text{Emb}(x_t)}{\|\text{Emb}(x_n)\|\|\text{Emb}(x_t)\|} \tag{4}$$

Based on the similarity scores, we filter out the top-$M$ similar examples, denoted as $Q$. For each prototype function $f_k$, we then compute its average performance score across the examples in $Q$:

$$s_{f_k} = \frac{1}{|Q|} \sum_{x_n \in Q} p_n^k \tag{5}$$

Here, $p_n^k$ denotes the average performance of $f_k$ on the example question $x_n$, while $s_{f_k}$ represents the mean accuracy across $Q$.

Rather than relying on pre-defined prompts, DPS actively learns which prototype functions yield the most effective demonstrations. By filtering out suboptimal prototypes, our method ensures that only the most informative relevant functions contribute to downstream reasoning, leading to more precise and adaptable prompt generation. The selected prototype functions are denoted as:

$$P = \{f_k, prompt_k \mid s_{f_k} > s_{LLM}\} \tag{6}$$

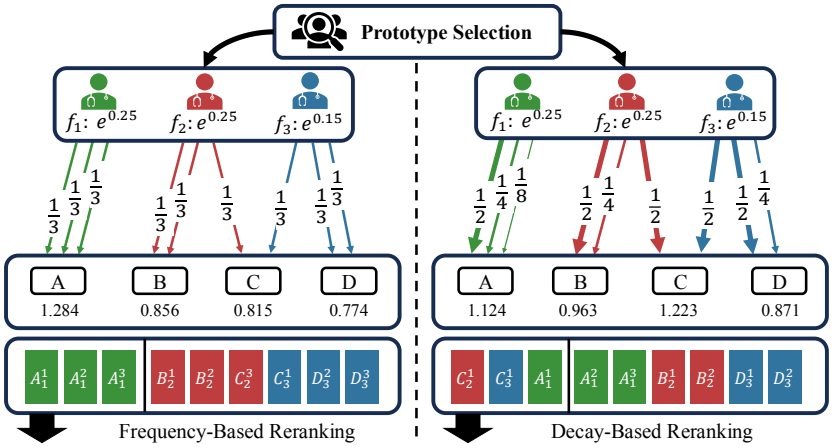



Fig. 3: Frequency-Based Reranking uses total voting frequency, while Decay-Based Reranking applies a decaying weight to each prototype function.

Here, $P$ represents the set of prototype functions $f_k$ whose scores $s_{f_k}$ are greater than the score of the downstream LLMs. The selected prototype functions serve as the foundation for the next stage, where they are leveraged to generate tailored demonstrations.

### 4.4 Prompt Synthesis

Traditional example selection methods rely on static retrieval and direct concatenation, limiting their adaptability to different queries. In contrast, DPS employs dynamic prototype selection and synthesis, allowing for adaptive and context-aware prompting. This ensures that the demonstrations are optimally tailored to the input query $x_t$t, significantly improving reasoning diversity and accuracy.

Specifically, when generating diverse reasoning paths, we can incorporate existing prompting methods to guide the prototype function, such as Task Instruction and Zero-shot CoT [12]. We would like to emphasize that previous prompt-based approaches can be integrated as part of our framework to enhance the quality of diverse reasoning paths. The impact of demonstration path quality on our method is discussed in detail in the analysis section. Formally, we prompt each function $f_k \in P$ to generate $D$ distinct demos for $x_t$ with prompt method $prompt_k$, denoted as:

$$f_k^d(x_t, prompt_k) = \left(\text{Exmp}_k^d, C_k^d\right), \tag{7}$$

where $\text{Exmp}_k^d$ represents the $d$-th response, and $C_k^d$ is the conclusion drawn from $\text{Exmp}_k^d$, ensuring that the generated demonstrations are both diverse and contextually relevant.

### 4.5 Voting-Based Reranking

Given the limited context window size, passing all $D \times |P|$ responses to the downstream LLMs is impractical. To retain the most relevant information, we propose a voting-based reranking method, scoring each $\text{Exmp}_k^d$ as $y_k^d$ by aggregating contributions from all prototype functions using a Softmax-like approach:

$$y_k^d = \sum_{f_k \in P} \exp(s_{f_k} - s_{LLM}) \cdot v_k^p, \tag{8}$$

where, $v_k^p$ represents the voting weight of $f_k$ for the $p$-th conclusion $C_p$ among all distinct conclusions $c_k^d$, and $\exp(s_{f_k} - s_{LLM})$ serves as a weighting factor that prioritizes high-performing prototype functions. The calculation of $v_k^p$ is performed using one of two methods:

***Frequency-based Reranking*** $v_k^p$ is assigned based on the frequency of each conclusion $C_p$ across all responses, as illustrated in Fig. 3.

***Decay-based Reranking*** To prevent a prototype function from dominating due to repeatedly producing the same conclusion, we apply an exponential decay strategy to the voting weights of each function's conclusions:

$$v_k^p = \frac{1}{D} \sum_{t=1}^{T_k^p} \left(2^{-t}\right) \tag{9}$$

Here, $T_k^p$ denotes the number of times $f_k$ produces the conclusion $C_p$. After voting-based reranking, we concatenate the top $H$ highest-scoring demonstrations to form $\text{Demo}(x_t, B)$. This ensures that only the most relevant and high-quality examples are passed to the LLM, maximizing contextual coherence and reasoning effectiveness.

## 5 Experimental Setup

### 5.1 Datasets

To evaluate *DPS*, we consider three reasoning tasks, each presenting unique challenges that necessitate diverse prototype selection.

- **Mathematical Reasoning** requires multi-step calculations and symbolic manipulations. This category includes five representative datasets: GSM8K [6] (GSM), MATH [9] (MTH), SVAMP [21] (SVA), ASDIV [18] (ASD), and MathQA [2] (MQA). These datasets allow us to examine how *DPS* improves problem-solving by leveraging multiple reasoning paths.
- **Commonsense Reasoning** involves understanding implicit world knowledge. Unlike mathematical reasoning, commonsense knowledge is often non-explicit and context-dependent. We use five benchmarks: CommonsenseQA [27] (CSQ), CommonsenseQA2 [28] (CS2), OpenbookQA [19] (OBQ), PIQA [3] (PIQ), and Com2Sense [26] (C2S). This task tests whether *DPS* can generate adaptive demonstrations that incorporate diverse commonsense perspectives.

– **Natural Language Inference (NLI)** requires determining logical relationships between sentence pairs. Compared to mathematical and commonsense reasoning, NLI involves recognizing entailment, contradiction, and neutrality in textual data. We evaluate on MNLI [33] (MLI), QNLI [30] (QLI), and ANLI [20] (ALI), focusing on whether *DPS* enhances reasoning robustness across different logical structures.

### 5.2 Baselines

We compare *DPS* with three baseline categories.

– **Basic Baselines.** Fundamental strategies with minimal guidance. **Prompting**: Task instructions without examples. **Random**: Randomly selecting examples from the bank. **CoT**: Prompting the model with "Think step by step."
– **Selection-based Methods.** Retrieving static examples from an example bank. **KNN**: Selecting the most similar examples via two embeddings: BGE [34] (bge-large-en-v1.5) [3]. Sentence-BERT [24] (all-MiniLM-L6-v2) [4]. **MMR** [37]: Balancing relevance and diversity. **ConE** [22]: Refining selection via LLM feedback. These methods retrieve relevant examples but remain **static**, which might limit adaptability in diverse tasks.
– **Generation-based Methods.** Dynamically constructing examples. **Analogy** [36]: Generating examples by recalling similar problems. **ComplexCoT** [7]: Constructing examples based on bank distributions. **AutoCoT** [39]: Automatically generating chain-of-thought demonstrations. These methods introduce adaptiveness but might not ensure **diversity and task alignment**, as they lack explicit quality control.

### 5.3 Implementation Details

*Dataset Usage.* We extract 500 samples from the official dataset for constructing the example bank and 1,000 samples for evaluation. For datasets with fewer than 1,500 samples, we retain 500 for the example bank and use the remainder for evaluation. If there is an official ground truth partition for the test or development sets, we directly adopt it. In the absence of an official partition, we perform random sampling to divide the dataset.

*Prototype Bank Construction.* We selected $N = 500$ examples to form the Prototype Bank. The candidate prototype functions used are `LLaMA3-8B-Instruct` [1], `MAmmoTH2-8B` [38] and `Apollo-7B` [32]. Our framework is flexible, allowing for the reuse of prototype functions and downstream models.

*Prototype Selection.* From the Prototype Bank, we select $M = 100$ most similar examples. For each prototype function, we sample 3 responses to calculate $p_n^k$. The generation temperature for all prototype functions was set to 0.7. For each model, we apply two prompting methods: Vanilla-Prompting and CoT-Prompting. This results in a total of six distinct prototype functions.

[3] https://huggingface.co/BAAI/bge-large-en-v1.5

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 1: Experimental results on Commonsense Reasoning benchmarks. The best results are highlighted in **bold**, and the second-best results are underlined.

| Category | Model | CSQ | OPQ | PIQ | CS2 | C2S | Avg. |
|---|---|---|---|---|---|---|---|
| Basic | Prompting | 71.4 | 71.6 | 79.4 | 62.6 | 62.0 | 69.4 |
| | Random | 72.6 | 74.2 | 77.2 | 63.8 | 66.5 | 70.8 |
| | CoT | 71.8 | 73.6 | 73.5 | 67.3 | 66.6 | 70.6 |
| Selection | KNN w/Mini [16] | 72.7 | 74.6 | 79.9 | 65.2 | 67.3 | 71.9 |
| | KNN w/BGE [16] | 72.9 | 76.6 | 80.1 | 65.0 | 66.6 | 72.2 |
| | MMR w/BGE [37] | 72.2 | 74.6 | 78.6 | 65.2 | 66.6 | 71.4 |
| | ConE [22] | 73.0 | 76.2 | 79.7 | 66.0 | 66.5 | 72.3 |
| Generation | Analogy [36] | 66.2 | 73.4 | 67.7 | 60.2 | 62.5 | 66.0 |
| | ComplexCoT [7] | 74.4 | 76.6 | 76.2 | 68.7 | **71.7** | 73.5 |
| | AutoCoT [39] | 75.8 | 75.4 | 76.4 | 68.9 | 71.5 | 73.6 |
| Ours | *frequency* | 77.9 | 80.0 | 81.8 | **71.7** | 68.7 | 76.1 |
| | *decay* | **79.2** | **80.6** | **82.3** | 71.6 | 69.9 | **76.7** |

*Prompt Synthesis.* Each selected prototype function generated $D = 4$ responses. Finally, the top $H = 4$ samples were selected for use in the demonstration. All baseline methods that provide examples follow a 4-shot setting to ensure a fair comparison. Due to computational resource constraints, each experiment was conducted once, and the performance results are reported accordingly. We use `LLaMA3-8B-Instruct` [1] as the downstream model. Al so, we report performance across various model types and sizes to provide a broader evaluation.

## 6 Main Results

### 6.1 Commonsense Reasoning Results

*DPS* achieves state-of-the-art accuracy on commonsense reasoning benchmarks, outperforming the strongest generative method, AutoCoT, by 3.1 points and the best selection-based model, Con [22], by 4.4 points (Table 1).

Notably, Analogy [36] exhibits subpar performance (66.0 avg.), significantly lower than basic prompting methods. We attribute this to its automatic generation of exemplars without validation, which often reinforces incorrect reasoning patterns. In contrast, other methods benefit from explicit or implicit example signals, leading to more stable performance improvements over prompting. This observation suggests that providing examples helps align large language models (LLMs) with human preferences.

Furthermore, in the PIQA benchmark, we observe that generative approaches perform markedly worse than selection-based methods. We hypothesize that this discrepancy arises because PIQA requires selecting the better option from two sentences, a task that relies more on human preference alignment than pure reasoning ability. In this context, *DPS* effectively balances preference exploitation and reasoning flexibility, thus surpassing all baselines.

Table 2: Results on Mathematical Reasoning benchmarks. The best results are highlighted in **bold**, and the second-best results are underlined.

| Model | GSM | MTH | SVA | ASD | MQA | Avg. |
|---|---|---|---|---|---|---|
| CoT | 75.4 | 29.9 | 84.8 | 79.7 | 53.3 | 64.6 |
| Analogy [36] | 60.0 | 27.0 | 75.0 | 78.6 | 51.1 | 58.3 |
| ComplexCoT [7] | 74.7 | 15.7 | 85.4 | 75.6 | 50.0 | 60.3 |
| AutoCoT [39] | 78.8 | 32.2 | 86.6 | 80.3 | 54.0 | 66.4 |
| *DPS-frequency* | 88.0 | 43.9 | **93.2** | 89.8 | 63.5 | 75.7 |
| *DPS-decay* | **88.2** | **45.4** | 93.0 | **89.9** | **66.0** | **76.5** |

### 6.2 Mathematical Reasoning Results

Table 2 shows that *DPS* achieves an average accuracy of 76.5, outperforming standard CoT (+11.9) and AutoCoT (+10.1) [39], highlighting the benefits of dynamic prototype selection and adaptive prompt synthesis in multi-step reasoning. Notably, ComplexCoT [7] underperforms compared to standard CoT, particularly on the MTH dataset. This may be due to its fixed reliance on the longest reasoning path, which often results in verbosity and error propagation. This observation suggests that heuristic rule-based exemplar selection tends to lack generalization capability. In comparison, *DPS* dynamically selects effective demonstrations and reduces redundancy through a voting-based mechanism, achieving the highest accuracy across all datasets.

In addition, the decay-based variant consistently achieves higher accuracy compared to the frequency-based variant. This suggests that decay-based reranking better prioritizes informative in-context demonstrations, reducing the influence of less relevant examples and thereby enhancing overall performance.

### 6.3 Natural Language Inference Results

*DPS* achieves the best overall performance on logical reasoning benchmarks, with *DPS-decay* attaining an average accuracy of 71.2, surpassing the strongest CoT-based method, AutoCoT, by 0.5 points and outperforming the best retrieval-based method, MMR w/BGE, by 9.5 points (Table 3).

Breaking down the results, selection-based approaches show modest improvements over the basic CoT model, but their effectiveness remains highly dependent on the quality of retrieved exemplars. In contrast, generation-based methods exhibit higher variance: ComplexCoT achieves strong performance in QLI (87.9) but suffers from severe instability in ALI and MLI, suggesting that its reliance on complexity-based heuristics leads to inconsistent reasoning across tasks.

In response to these limitations, *DPS* dynamically selects and ranks high-quality demonstrations, ensuring stable performance across datasets. Notably, its frequency-based variant already outperforms all baselines, while the decay-based strategy further refines example weighting, leading to improved robustness and adaptability. These findings underscore *DPS* as an effective and generalizable approach for in-context learning in logical reasoning tasks.

Table 3: Results on Logical Reasoning benchmarks. The best results are highlighted in **bold**, and the second-best results are underlined.

| Category | Model | ALI | MLI | QLI | Avg. |
|---|---|---|---|---|---|
| Basic | Prompting | 57.9 | 59.7 | 58.3 | 58.6 |
| | Random | 61.0 | 59.7 | 58.8 | 61.5 |
| | CoT | 67.2 | 61.7 | 78.0 | 69.0 |
| Selection | KNN w/Mini [16] | 61.4 | 61.7 | 60.0 | 61.2 |
| | KNN w/BGE [16] | 61.5 | 61.8 | 57.7 | 61.0 |
| | MMR w/BGE [37] | 61.9 | 62.0 | 58.1 | 61.7 |
| | ConE [22] | 63.3 | 62.0 | 59.2 | 61.5 |
| Generation | Self-Gen [36] | 42.4 | 39.8 | 43.4 | 41.9 |
| | ComplexCoT [7] | 21.6 | 16.1 | **87.9** | 41.9 |
| | AutoCoT [39] | 70.4 | 61.4 | 80.3 | 70.7 |
| DPS (Ours) | *frequency* | 71.1 | 61.9 | 79.7 | 70.9 |
| | *decay* | **71.5** | **62.1** | 80.0 | **71.2** |

Table 4: Ablation study of *DPS* (frequency-based reranking) on six datasets.

| Method | GSM | MTH | SVA | ASD | CSQ | CS2 | Avg. |
|---|---|---|---|---|---|---|---|
| ***DPS*** | **88.0** | **43.9** | **93.2** | **89.8** | **77.9** | **71.7** | **77.4** |
| **w/o Prototype Selection** | 86.6 | 28.0 | 89.4 | 86.8 | 74.4 | 66.7 | 72.0 |
| **w/o Prompt Synthesis** | 72.8 | 28.7 | 84.8 | 79.6 | 76.0 | 70.3 | 68.7 |
| **w/o Demo Reranking** | 81.5 | 37.4 | 85.8 | 81.1 | 77.3 | 70.9 | 72.3 |

## 7 Analysis

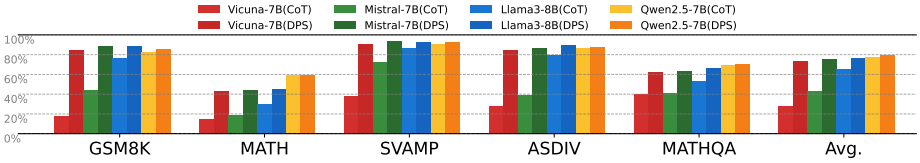
*The contribution of different components.* We test *DPS* in three settings (Table 4). **w/o Prototype Selection**: prototypes are randomly selected; **w/o Prompt Synthesis**: only final answers are sampled from prototypes, excluding the chain of thought; and **w/o Demo Ranking**: demonstrations are presented in random order. The results show that each component is crucial for high-quality demonstrations, with the largest performance drop occurring when Prototype Selection is removed, emphasizing its importance in *DPS*.

*Cost-Effectiveness Analysis.* During Prototype Selection, *DPS* performs KNN retrieval over a relatively small text corpus, resulting in negligible latency. During Prompt Synthesis, *DPS* employs multiple sampling from prototype functions. Thus we compare it with the self-consistency approach under various settings. Table 5 shows that *DPS* consistently outperforms self-consistency across all task types while utilizing only 15% of the sampling budget. This advantage arises from *DPS*'s ability to generate diverse reasoning paths from multiple perspectives, thereby introducing greater variability compared to self-consistency.

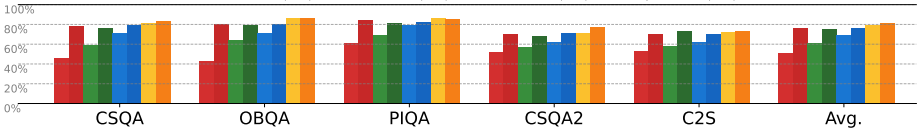
*Comparison of Models and Sizes.* We evaluated *DPS* on four model families: Vicuna [5], Mistral [11], Llama3 [1], and Qwen2.5 [29]. Fig. 4 shows average

Table 5: Comparison of *DPS* and Self-Consistency methods across datasets. The best results are highlighted in **bold**, and the second-best results are underlined.

| | **SC-Vanilla** | | **SC-CoT** | | **D-freq** | **D-decay** |
|---|---|---|---|---|---|---|
| #sample | 10 | 40 | 10 | 40 | 6 | 6 |
| SVAMP | 69.8 | 70.2 | 89.2 | 91.4 | **93.2** | 93.0 |
| ASDIV | 64.9 | 65.3 | 84.7 | 85.4 | 89.8 | **89.9** |
| CSQA | 71.2 | 71.2 | 73.1 | 74.2 | 77.9 | **79.2** |
| CSQA2 | 62.4 | 62.9 | 68.9 | 69.2 | **71.7** | 71.6 |




(a) Mathematical Reasoning.




(b) Commonsense Reasoning.

Fig. 4: Accuracy of *DPS* across four different downstream LLMs on various reasoning tasks. The results demonstrate that the DPS framework brings significant performance improvements to the different downstream LLMs.

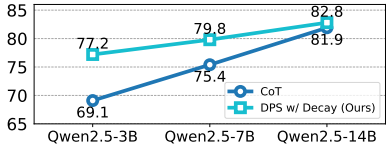



Fig. 5: The effect of different model sizes on reasoning performance is reported with the average accuracy across datasets of the Qwen2.5 family.

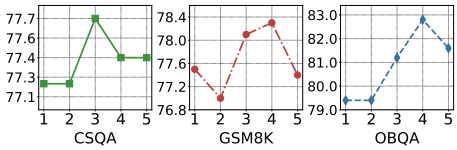



Fig. 6: Accuracy trends with different $H$ values. Accuracy improves with increasing $H$ up to a certain threshold, beyond which performance declines.

accuracy across all datasets, highlighting that models with lower baseline performance benefit most from *DPS*. For example, Vicuna-7B improved by 35.5%, while Qwen2.5-7B gained 4.4% despite a strong baseline. Fig. 5 shows that larger models also benefit, though smaller models see larger gains.

*Number of Demonstrations.* Fig. 6 reveals the following insights: 1) As more demonstrations are introduced, the performance of *DPS* improves, indicating that a single example is insufficient. 2) However, when the number of demonstrations becomes too large, additional examples provide redundant information, leading to a decline in performance. Based on these findings, we recommend using three demonstration examples and suggest adjusting this number for optimal performance depending on the specific dataset.

## 8 Conclusion

We introduce *DPS*, a framework that enhances the example bank through diverse prototype selection. *DPS* decouples in-context learning into two stages: Prototype Selection, where diverse prototype functions are chosen, and Prompt Synthesis, where these functions generate demonstrations for the input query. By incorporating voting-based reranking, *DPS* introduces high-quality demonstrations to the downstream LLM. Extensive experiments across 13 benchmarks in three reasoning domains show that *DPS* consistently improves model performance, highlighting its effectiveness in refining in-context learning.

Our work has several potential limitations. While our framework does not face significant computational budget constraints, the example selection process does introduce some inference latency, which could become a concern for very large models where inference speed is critical. Future work could explore more efficient strategies for example selection to reduce this latency and improve scalability.

**Acknowledgments.** This work is supported in part by Ucap Cloud and the State Key Laboratory of General Artificial Intelligence.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
2. Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., Hajishirzi, H.: MathQA: Towards interpretable math word problem solving with operation-based formalisms
3. Bisk, Y., Zellers, R., Bras, R.L., Gao, J., Choi, Y.: Piqa: Reasoning about physical commonsense in natural language (2019)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. pp. 1877–1901 (2020)
5. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (2023)

6. Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J.: Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 (2021)
7. Fu, Y., Peng, H., Sabharwal, A., Clark, P., Khot, T.: Complexity-based prompting for multi-step reasoning (2023)
8. Guo, Q., Wang, L., Wang, Y., Ye, W., Zhang, S.: What makes a good order of examples in in-context learning. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics: ACL 2024
9. Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., Steinhardt, J.: Measuring mathematical problem solving with the math dataset. NeurIPS (2021)
10. Hosseini, P., Castro, I., Ghinassi, I., Purver, M.: Efficient solutions for an intriguing failure of LLMs: Long context window does not mean LLMs can analyze long sequences flawlessly. In: Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S. (eds.) Proceedings of the 31st International Conference on Computational Linguistics. Association for Computational Linguistics, Abu Dhabi, UAE (2025)
11. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
12. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022)
13. Lee, J., Yang, W., Gupta, G., Wei, S.: Automatic mathematic in-context example generation for LLM using multi-modal consistency. In: Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S. (eds.) Proceedings of the 31st International Conference on Computational Linguistics. Association for Computational Linguistics, Abu Dhabi, UAE (2025)
14. Li, M., Liu, Z., Deng, S., Joty, S., Chen, N., Kan, M.Y.: DnA-eval: Enhancing large language model evaluation through decomposition and aggregation. In: Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S. (eds.) Proceedings of the 31st International Conference on Computational Linguistics. Association for Computational Linguistics, Abu Dhabi, UAE (2025)
15. Liu, H., Liu, J., Huang, S., Zhan, Y., Sun, H., Deng, W., Wei, F., Zhang, Q.: $se^2$: Sequential example selection for in-context learning. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics: ACL 2024. Association for Computational Linguistics, Bangkok, Thailand (2024)
16. Liu, J., Shen, D., Zhang, Y., Dolan, W.B., Carin, L., Chen, W.: What makes good in-context examples for gpt-3? In: DeeLIO (2022)
17. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024)
18. Miao, S.y., Liang, C.C., Su, K.Y.: A diverse corpus for evaluating and developing english math word problem solvers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 975–984 (2020)
19. Mihaylov, T., Clark, P., Khot, T., Sabharwal, A.: Can a suit of armor conduct electricity? a new dataset for open book question answering. In: Conference on Empirical Methods in Natural Language Processing (2018)
20. Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., Kiela, D.: Adversarial nli: A new benchmark for natural language understanding. In: ACL (2020)

21. Patel, A., Bhattamishra, S., Goyal, N.: Are NLP models really able to solve simple math word problems? In: NAACL. Online (2021)
22. Peng, K., Ding, L., Yuan, Y., Liu, X., Zhang, M., Ouyang, Y., Tao, D.: Revisiting demonstration selection strategies in in-context learning. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand (2024)
23. Peng, K., Ding, L., Zhong, Q., Shen, L., Liu, X., Zhang, M., Ouyang, Y., Tao, D.: Towards making the most of chatgpt for machine translation. In: Findings of EMNLP (2023)
24. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2019)
25. Ren, Z., Zhan, Y., Yu, B., Ding, L., Tao, D.: Healthcare copilot: Eliciting the power of general llms for medical consultation. arXiv preprint (2024), https://arxiv.org/abs/2402.13408
26. Singh, S., Wen, N., Hou, Y., Alipoormolabashi, P., Wu, T.l., Ma, X., Peng, N.: COM2SENSE: A commonsense reasoning benchmark with complementary sentences. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021
27. Talmor, A., Herzig, J., Lourie, N., Berant, J.: CommonsenseQA: A question answering challenge targeting commonsense knowledge. In: NAACL. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
28. Talmor, A., Yoran, O., Bras, R.L., Bhagavatula, C., Goldberg, Y., Choi, Y., Berant, J.: Commonsenseqa 2.0: Exposing the limits of ai through gamification (2022)
29. Team, Q.: Qwen2.5: A party of foundation models (2024)
30. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
31. Wang, S., Ding, L., Shen, L., Luo, Y., Du, B., Tao, D.: Oop: Object-oriented programming evaluation benchmark for large language models. arXiv preprint (2024), https://arxiv.org/abs/2401.06628
32. Wang, X., Chen, N., Chen, J., Hu, Y., Wang, Y., Wu, X., Gao, A., Wan, X., Li, H., Wang, B.: Apollo: Lightweight multilingual medical llms towards democratizing medical ai to 6b people (2024)
33. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics (2018)
34. Xiao, S., Liu, Z., Zhang, P., Muennighoff, N.: C-pack: Packaged resources to advance general chinese embedding (2023)
35. Xu, Z., Peng, K., Ding, L., Tao, D., Lu, X.: Take care of your prompt bias! investigating and mitigating prompt bias in factual knowledge extraction. In: LREC-COLING (2024)
36. Yasunaga, M., Chen, X., Li, Y., Pasupat, P., Leskovec, J., Liang, P., Chi, E.H., Zhou, D.: Large language models as analogical reasoners. In: The Twelfth International Conference on Learning Representations (2024)
37. Ye, X., Iyer, S., Celikyilmaz, A., Stoyanov, V., Durrett, G., Pasunuru, R.: Complementary explanations for effective in-context learning. In: Rogers, A., Boyd-Graber,

J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics: ACL 2023. Association for Computational Linguistics, Toronto, Canada (2023)
38. Yue, X., Zheng, T., Zhang, G., Chen, W.: Mammoth2: Scaling instructions from the web. arXiv preprint arXiv:2405.03548 (2024)
39. Zhang, Z., Zhang, A., Li, M., Smola, A.: Automatic chain of thought prompting in large language models. In: ICLR (2023)