Correlation-Aware Example Selection for In-Context Learning with Nonsymmetric Determinantal Point Processes

Anonymous ACL submission

Abstract

LLMs with in-context learning (ICL) obtain remarkable performance but are sensitive to the quality of ICL examples. Prior work on ICL example selection explored unsupervised heuristic methods and supervised LLM feedback-based methods, but they typically focus on the selection of individual examples, ignore correlations among examples. Recent researchers propose to use the determinantal point process (DPP) to model negative correlations among examples to select diverse example sets. However, the DPP fails to model positive correlations among examples, but ICL still requires the positive correlations of examples to ensure the consistency of its examples that provide a clear instruction for LLMs. In this paper, we propose an ICL example selection framework based on the nonsymmetric determinantal point process (NDPP) to capture positive and negative correlations, consider both the diversity and the relevance among ICL examples. Specifically, we optimize NDPP via kernel decomposition-based MLE to fit a constructed pseudo-labeled dataset, where we also propose low-rank decomposition to reduce the computational cost. Further, we perform query-aware kernel adaptation on our NDPP to customize the input query, and we select examples via a maximal-a-posteriori inference based on the adapted NDPP. Experiments show our model excels strong baselines in ICL example selection.

1 Introduction

Large language models (LLMs) show good performance through in-context learning (ICL) (Brown et al., 2020; Wei et al., 2022b,a; Wen et al., 2024; Pan et al., 2024). ICL typically uses an example set and a task-specific instruction as a prompt and inputs a concatenation of the prompt and an user's input query into LLMs. ICL allows LLMs to perform tasks by observing a series of examples without the need to update parameters. However, the performance of ICL is sensitive to the selection of examples (Liu et al., 2022; Zhang et al., 2022; Min et al., 2022; An et al., 2023). Recent works (Lu et al., 2022; Cheng et al., 2023) also show that different example sets exhibit significant differences in performance. Thus, example selection is crucial for exploiting the ICL capabilities of LLMs.

To select suitable examples for ICL, researchers propose various context-dependent heuristic methods, where they select examples according to examples' entropy (Lu et al., 2022), complexity (Fu et al., 2022), perplexity (Gonen et al., 2023), and diversity (Li and Qiu, 2023). These methods outperform random selection, but these methods ignore characteristics of the specific input queries and thus cannot customize the ICL example set for the input queries. To consider the query, researchers propose context-aware methods to retrieve similar examples for ICL (Liu et al., 2022; Agrawal et al., 2023; Hongjin et al., 2022). They use off-the-shelf retrievers such as BM25 (Robertson et al., 2009) or SBERT (Reimers and Gurevych, 2019) to select examples based on their textual or semantic similarity to the query. When applying LLMs to specific tasks, they cannot customize the example selection of ICL for the given task since the ICL example selector (i.e., retriever) is not learnable and cannot learn to tailor for the task-specific data.

To leverage task supervision, some recent work (Rubin et al., 2022; Cheng et al., 2023; Li et al., 2023; Xiong et al., 2024) introduce LLMs feedback as the task-specific supervisory signal to train the ICL example selectors (i.e. retriever), where the signal is used to rank and label examples. In these methods, the retrievers learn the LLMs' preference for examples in different tasks, and adaptively select examples for each task. However, they typically focus on the selection of each individual example, ignore the correlations (i.e., interrelationships) among a set of ICL examples.

To consider the correlations among examples for

ICL, researchers (Levy et al., 2023; Ye et al., 2023a; Yang et al., 2023) propose to use the determinantal point process (DPP) (Kulesza and Taskar, 2012) to select examples by balancing the *relevance to input queries* and the *diversity among examples*. They model the relevance to input queries by similarity between queries and examples, and they model the diversity among examples since DPP's kernel matrix $\boldsymbol{L}$ models the negative correlation of data points. However, DPP's kernel matrix $\boldsymbol{L}$ is a symmetric positive semi-definite (PSD) matrix. $\boldsymbol{L}$ restricts DPP can only model negative correlation [1] among examples rather than positive correlation. It results in DPP ignoring the *relevance among candidate examples*.

We argue the ICL example selection should not only consider the *relevance to input queries* and the *diversity among examples*, but also cater to the *relevance among examples*. Ensuring the consistency of ICL examples contributes to providing a clear instruction to guide the LLMs (Liu et al., 2024a). [2]

In this paper, we propose an ICL example selection method for LLM based on the nonsymmetric determinantal point process model (NDPP), which considers the *relevance to input queries*, the *diversity among ICL examples*, and the *relevance among ICL examples*. NDPP's nonsymmetric property makes the selection model relevance among ICL examples. Specifically, we construct an NDPP model with a kernel matrix to capture positive and negative correlations among ICL examples. In the training stage, we propose a kernel decomposition-based maximum likelihood estimation (KD-MLE) to train the NDPP by fitting the kernel matrix over our constructed pseudo-labeled datasets. To reduce the computational cost of KD-MLE, we propose a low-rank decomposition of the kernel matrix. In the inference stage, to consider the *relevance to input queries*, we propose a query-aware kernel adaptation, which adapts the trained NDPP to the given query by incorporating the embedding similarity between examples and queries into the kernel matrix. We finally perform maximal-a-posteriori (MAP) inference based on the adapted NDPP to select the ICL example set for LLMs. Experiments show that our method exceeds baselines on five datasets, including open-domain QA, code generation, semantic parsing and story generation tasks. Our code is released.[3]

Our contributions are: (1) We propose a novel ICL example selection framework based on NDPP, which captures positive and negative correlations among examples and learns the composition of ICL examples to select suitable ICL examples for LLM. (2) We propose a query-aware kernel optimization to consider the similarity between queries and examples, which enables our framework to select customized ICL example sets for different queries. (3) Experiments on five datasets show that our method achieves SOTA on ICL example selection.

2 Related Work

2.1 Example Selection for ICL

The in-context learning (ICL) performance of LLMs depends on the selection of examples. Depending on whether the query information and the task supervision were considered, ICL example selection methods can be divided into three categories: (1) **In-context Insensitive Unsupervised Methods.** These approaches ignore the query information and task supervision. Fu et al. (2022) propose a complexity-based example selection method. Lu et al. (2022) Propose an entropy-based approach to mitigate example order sensitivity. Li and Qiu (2023) use a diversity-guided example search strategy to select examples. (2) **In-context Sensitive Unsupervised Methods.** This category considers query information but ignores the task supervision. Researchers find that selecting different examples can reduce the redundancy of ICL example set (Liu et al., 2022; Agrawal et al., 2023; Hongjin et al., 2022). Wang et al. (2024a) further propose a model-specific example selection method based on feature evaluation to improve ICL performance during inference. Similarly, Liu et al. (2024b) select examples with multiple levels of similarity to queries to improve ICL performance. (3) **In-context Sensitive Supervised Methods.** By introducing task supervision, these methods fine-tune the ICL example selector (i.e. retriever) for more precise example selection. Many studies have improved the quality of ICL examples by iteratively training retrievers (Rubin et al., 2022; Wang et al., 2024b; Li et al., 2023; Liu et al., 2024b). Besides, Xiong et al.

[1]In DPP, the correlation between examples i and j is expressed as $-\boldsymbol{L}_{ij}\boldsymbol{L}_{ji}$, where $\boldsymbol{L}$ is the kernel matrix. Due to the symmetric property of PSD matrix, $\boldsymbol{L}_{ij}$ and $\boldsymbol{L}_{ji}$ are always equal, making the correlation $-\boldsymbol{L}_{ij}\boldsymbol{L}_{ji}$ always non-positive.

[2]The relevance and diversity are not conflicted since ICL needs multiple examples, where some of them may be diverse and others are relevant so as to provide a comprehensive and consistent instruction to LLMs.

[3]anonymous.4open.science/r/ICL_example_selection_with_NDPP-FE36

(2024) use chain-of-thought generated by LLMs to refine the retriever. Fu et al. (2022) propose to optimize the retriever by calculating semantic similarity, example diversity, and event correlation. To consider diversity among examples, Levy et al. (2023); Yang et al. (2023); Ye et al. (2023b) employ DPP to select diverse example sets. These works only consider relevance to input queries and diversity among examples, our framework further considers relevance among examples.

2.2 Determinantal Point Processes and Its Applications

Determinantal Point Process (DPP) is a probabilistic model that can select diverse subsets by capturing negative correlations among items of the set.

DPP has seen significant development. Johansson et al. (2023) proposed a semi-supervised k-DPP method. Grosse et al. (2024) used a greedy algorithm for k-DPP sampling. To reduce computational complexity, more efficient inference methods were proposed, such as LSMOEA-DPP (Okoth et al., 2022) and Anisotropic DPP (Ghilotti et al., 2024).

DPP is widely used in AI applications, especially for tasks that require diversity sets, such as neural network training (Sheikh et al., 2022), recommendation systems (Liu et al., 2024c), video analysis (Chen et al., 2023), and abstract summary (Shen et al., 2023). DPP also been used to optimize GNN on graph-structured data. (Duan et al., 2022).

Gartrell et al. (2019) propose an extension of DPP called nonsymmetric determinantal point processes (NDPP), which can model both positive and negative correlations among a set of items. Gartrell et al. (2021) reduce NDPP's complexity via kernel decomposition. Han et al. (2022) propose a scalable sampling method for NDPP. Song et al. (2024) propose a fast dynamic algorithm for resampling distributions of NDPP, which shortens the sampling time.

While current works focus on the application of the DPP, we explore the application of the NDPP on ICL example selection.

3 Preliminary

In-Context Learning. In-context learning (ICL) (Brown et al., 2020) prompts are usually sequences of examples. Given test instance (x_{test}, y_{test}), LLMs predicts $\hat{y}$ with k-shot ICL prompt :

$$\hat{y} = LLM(e_1 \oplus, ..., \oplus e_k \oplus x_{test}) \quad (1)$$

Where $e_i = (x_i, y_i)_{i=1}^k$ is the i_{th} example, and $\oplus$ is the concatenation operation. The objective of ICL example selection task is to select k examples from a pre-constructed example pool such that the predicted value $\hat{y}$ matches its ground truth y_{test}.

Nonsymmetric Determinantal Point Process. Nonsymmetric determinantal point process (NDPP) is a probabilistic model to model correlations between items in a set (Gartrell et al., 2019). It models a finite ground set D with a kernel matrix $\boldsymbol{L}$ such that for any subset $E \in D$, $Pr(E) \propto det(\boldsymbol{L}_E)$, where $\boldsymbol{L}_E$ is the submatrix of $\boldsymbol{L}$ indexed by E .Given the kernel matrix $\boldsymbol{L}$, the probability a subset E being selected from D is defined as:

$$P_{\boldsymbol{L}}(E) = \frac{det(\boldsymbol{L}_E)}{det(\boldsymbol{L} + I)} \quad (2)$$

where I is the unit matrix.

4 Method

4.1 Overview

To provide high-quality ICL examples for LLMs, we construct an ICL example selection framework based on the NDPP model, where the NDPP consists of a kernel matrix $\boldsymbol{L}$ to model correlations among examples. We construct a pseudo-labeled training set based on LLMs feedback (§ 4.2), and use the pseudo-labeled training set to train the NDPP model by kernel decomposition-based maximum likelihood estimation (KD-MLE) (§ 4.3). In the inference stage, we perform query-aware kernel-adaptation on the trained NDPP model to consider the relevance to input queries, and select ICL examples based on the adapted model through MAP inference (§ 4.4).

4.2 Example Subsets Pseudo-labeling via LLMs' Feedback

Since there is no ground truth of ICL example sets for each training instance, to train the NDPP model in § 4.3 by MLE, we collect the feedback signals from LLMs for scoring the example subsets to construct a training set.

Given a task, we construct the pseudo-labeled training set with three steps: (1) **Candidate example retrieval.** For each instance (x_i, y_i) from our training set, we retrieve a candidate example set from the example pool D using the KNN retriever, which considers the embedding similarity between

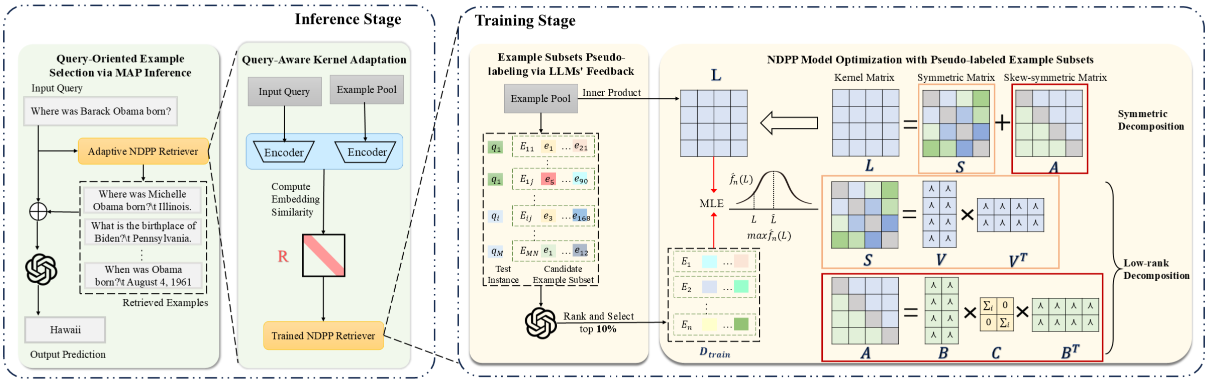

Figure 1: The overview of our framework. In the training stage, we construct a pseudo-labeled training set D_{train} based on LLMs' feedback (§ 4.2), and use D_{train} to optimize the kernel matrix $\boldsymbol{L}$ of the NDPP model by kernel decomposition-based MLE (§ 4.3). In the inference stage, we perform query-aware kernel-adaptation on the trained NDPP model, and select ICL examples based on the adapted model through MAP inference (§ 4.4).

the instance and examples. From the retrieved candidate example set, we randomly sample N non-overlapping subsets, denoted as $\{E_{ij}\}_{j=1}^{N}$. (2) **Example subset scoring.** We measure the quality of each candidate example subset E_{ij} with a quality score s_{ij}, and the scores act as pseudo labels of the subsets. To obtain the quality score s_{ij}, we concatenate the query x_i and examples in the subset E_{ij}, and input the concatenation into an LLM to obtain the probability $P_{LLM}(y_i|E_{ij}, x_i)$ of predicting the corresponding ground truth y_i of the test query x_i, which is formalized as: $s_{ij} = P_{LLM}(y_i|E_{ij}, x_i)$. (3) **Pseudo training set construction.** We rank candidate example subsets based on the score s_{ij}, and select the top 10% high-scoring subsets for all instances to construct a pseudo-labeled training set $D_{train} = (E_i)_{i=1}^{n}$, where n is the subset number. D_{train} is used to train the NDPP model in (§ 4.3).

4.3 NDPP Model Optimization with Pseudo-labeled Example Subsets

To select high-quality ICL example sets, we train the NDPP model by kernel decomposition-based maximum likelihood estimation (KD-MLE), which allows the NDPP model to learn the kernel matrix of high-scoring example subsets from the pseudo-labeled training set. The process consists of three steps: (1) we first define the NDPP optimization objective, then (2) get the kernel decomposition for NDPP, and finally, (3) we optimize NDPP via the kernel decomposition-based MLE.

4.3.1 NDPP Optimization Objective: MLE with Kernal Matrix

To capture correlations among examples in the ICL example set, we optimize the kernel matrix of the ICL example set to fit the pseudo-labeled training set. The fitted kernel matrix represents the feature of high-scoring ICL example sets so that the NDPP model can select suitable examples with the fitted kernel matrix.

In the NDPP, recall that the probability of selecting a candidate example subset E_i from the example pool D is $P_{\boldsymbol{L}}(E_i) = \frac{det(\boldsymbol{L}_{E_i})}{det(\boldsymbol{L}+\boldsymbol{I})}$ (as shown in Eq. 2), where $\boldsymbol{L}$ is the kernel matrix of D and $\boldsymbol{L}_{E_i}$ is the submatrix of $\boldsymbol{L}$ indexed by E_i. The base kernel matrix $\boldsymbol{L}$ is constructed by computing the pairwise embedding similarity between two examples $\langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle$ in the example pool D, where $\boldsymbol{L}_{ij} = sim(\boldsymbol{e}_i, \boldsymbol{e}_j)$. Elements of $\boldsymbol{L}$ show correlations among examples in the example pool. Given different kernel matrices, the NDPP selects different ICL example sets with the probability $P_{\boldsymbol{L}}(\cdot)$.

To select high-quality ICL example sets with NDPP, we aim to find a kernel matrix $\boldsymbol{L}$ that maximizes the probability of selecting high-scoring ICL example subsets. To achieve it, we optimize the kernel matrix $\boldsymbol{L}$ of the ICL example set to fit the pseudo-labeled training set $D_{train} = (E_i)_{i=1}^{n}$. Specifically, we optimize $\boldsymbol{L}$ towards the log-likelihood on the training set D_{train} as,

$$\hat{f}_n(L) = \frac{1}{n}\sum_{i=1}^{n} logP_L(E_i) \tag{3}$$

Because $P_{\boldsymbol{L}}(E_i) = \frac{det(\boldsymbol{L}_{E_i})}{det(\boldsymbol{L}+\boldsymbol{I})}$, we have:

$$\hat{f}_n(\boldsymbol{L}) = \frac{1}{n}\sum_{i=1}^{n} logdet(\boldsymbol{L}_{E_i}) - logdet(\boldsymbol{L}+\boldsymbol{I}) \tag{4}$$

The optimized kernel matrix $\hat{\boldsymbol{L}}$ is the kernel matrix that maximizes the Eq. 4, denoted as:

$$\hat{\boldsymbol{L}} = \arg\max_{\boldsymbol{L}} \hat{f}_n(\boldsymbol{L}) \tag{5}$$

The optimized kernel matrix $\hat{\boldsymbol{L}}$ is the learnable optimal approximation of high-scoring ICL example subsets' kernel matrix, with its elements representing correlations among examples.

4.3.2 Kernel Decomposition of NDPP

To optimize the kernel matrix $\boldsymbol{L}$ conveniently, we perform a two-step decomposition on the NDPP kernel matrix: we first perform symmetric decomposition on the kernel matrix, which enables NDPP to learn the positive and negative correlations among examples independently, and then perform a low-rank decomposition to reduce the computational cost. Details are as follows:

Symmetric decomposition. To distinguish the positive and negative correlations among examples (using NDPP's nonsymmetric property), we decompose the kernel matrix $\boldsymbol{L}$ into the sum of a symmetric matrix $\boldsymbol{S}$ and a skew-symmetric matrix $\boldsymbol{A}$ as in Eq. 6, where $\boldsymbol{A}$ and $\boldsymbol{S}$ denote the positive and negative correlations, respectively.

Low-rank decomposition. To reduce the computational cost, inspired by Gartrell et al. (2021), we further perform a low-rank decomposition on the symmetric matrix $\boldsymbol{S}$ and the skew-symmetric matrix $\boldsymbol{A}$ as in Eq. 6, which converts the high-dimensional representation of the correlations into a low-dimensional representation.

$$\boldsymbol{L} = \boldsymbol{S} + \boldsymbol{A}, \boldsymbol{S} = \boldsymbol{V}\boldsymbol{V}^T, \boldsymbol{A} = \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T \tag{6}$$

$\boldsymbol{V}, \boldsymbol{B} \in \mathbb{R}^{M\times K}$ are low-rank matrices of $\boldsymbol{S}$ and $\boldsymbol{A}$ respectively, where M is the example number in the example pool D and K is the rank of the kernel matrix $\boldsymbol{L}$. $\boldsymbol{V}$ and $\boldsymbol{B}$ indicate the low-dimensional representation of the negative and positive correlations among examples, respectively. $\boldsymbol{C} \in \mathbb{R}^{K\times K}$ is a block-diagonal matrix with diagonal blocks Σ_i of the form $\begin{bmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{bmatrix}$, where $\lambda_i > 0$. $\boldsymbol{C}$ maintains the skew-symmetric property of $\boldsymbol{A}$.

4.3.3 Kernel Decomposition-based MLE

We perform MLE to fit the kernel matrix $\boldsymbol{L}$ with its kernel decomposition form $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T$ obtained in the above step, where we also apply a regularization term to the log-likelihood.

Step 1: Kernel-decomposed MLE. When we optimize the kernel matrix $\boldsymbol{L}$ towards the MLE objective, we need to perform the decomposition of $\boldsymbol{L}$ to ensure that $\boldsymbol{L}$ captures both positive and negative correlations. We recall that the log-likelihood of the kernel matrix $\boldsymbol{L}$ (Eq. 4). Specifically, we use the decomposition form $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T$ in Eq. 6 to decompose $\boldsymbol{L}$ and $\boldsymbol{L}_{E_i}$ in the objective function (Eq. 4) to obtain the kernel-decomposed log-likelihood (Eq. 7),

$$\begin{aligned} &\phi(\boldsymbol{V}, \boldsymbol{B}, \boldsymbol{C}) \\ &= \frac{1}{n}\sum_{i=1}^{n} logdet\left(\boldsymbol{V}_{E_i}\boldsymbol{V}_{E_i}{}^T + \boldsymbol{B}_{E_i}\boldsymbol{C}\boldsymbol{B}_{E_i}{}^T\right) \\ &- logdet\left(\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T + \boldsymbol{I}\right) \end{aligned} \tag{7}$$

Eq. 7 allows us to optimize the log-likelihood with the decomposed components $\boldsymbol{V}$, $\boldsymbol{B}$, $\boldsymbol{C}$. The matrices $\boldsymbol{B}$ and $\boldsymbol{V}$ can capture positive and negative correlations among examples respectively.

Step 2: Regularized log-likelihood. To prevent overfitting, we define a regularization term as shown in Eq. 8. We perform L2 regularization for each row vector $\boldsymbol{v}_i$ and $\boldsymbol{b}_i$ of the matrices $\boldsymbol{V}$ and $\boldsymbol{B}$ separately, and use hyperparameters α and β to control the regularization strength of the matrices $\boldsymbol{V}$ and $\boldsymbol{B}$, respectively. In addition, we define a weight parameter $\frac{1}{\gamma_i}$ to control the regularization strength for each row vector, where γ_i denotes the occurrences of the i_{th} element appears in D_{train}. The regularization term is formally denoted as:

$$R(\boldsymbol{V}, \boldsymbol{B}) = -\alpha\sum_{i=1}^{M}\frac{1}{\gamma_i} \parallel \boldsymbol{v}_i \parallel_2^2 - \beta\sum_{i=1}^{M}\frac{1}{\gamma_i} \parallel \boldsymbol{b}_i \parallel_2^2 \tag{8}$$

Adding the regularization term (Eq. 8) to the kernel-decomposed log-likelihood (Eq. 7), we obtain the regularized log-likelihood (Eq. 9):

$$\begin{aligned} &\phi(\boldsymbol{V}, \boldsymbol{B}, \boldsymbol{C}) \\ &= \frac{1}{n}\sum_{i=1}^{n} logdet\left(\boldsymbol{V}_{E_i}\boldsymbol{V}_{E_i}{}^T + \boldsymbol{B}_{E_i}\boldsymbol{C}\boldsymbol{B}_{E_i}{}^T\right) \\ &- logdet\left(\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^T + I\right) \\ &+ R(\boldsymbol{V}, \boldsymbol{B}) \end{aligned} \tag{9}$$

In summary of the processing of § 4.3, we first train the NDPP model on the pseudo-labeled training set D_{train} collected in § 4.2, where we optimize Eq. 9 to find the optimized kernel matrix (§ 4.3.1) $\hat{\boldsymbol{L}}$ through its kernel decomposition form (§ 4.3.2 and§ 4.3.3) as Eq. 6. Then, the optimized kernel matrix can assist the NDPP model to select high-quality ICL example sets.

4.4 ICL Example Selection via NDPP for LLMs Inference

In the inference stage, to provide customized high-quality ICL examples for different queries, we propose query-aware kernel adaptation to adapt the trained NDPP to specific input queries so as to select ICL examples. To achieve it, we adapt the NDPP to input queries by modeling the similarity between examples and queries (§ 4.4.1), and then select ICL examples by maximum-a-posteriori (MAP) inference using the adapted NDPP (§ 4.4.2). The above operations consider both the relevance to input queries and the relevance among examples.

4.4.1 Adapting NDPP to Input Queries

To adapt the NDPP to input queries, we update the kernel matrix of NDPP by introducing the similarity between examples and input queries into the kernel matrix.

For each query, we update that kernel matrix with three steps: (1) **Similarity Score Computation.** We encode the query x via a query encoder $E_Q(\cdot)$ and encode the example e_i via an example encoder $E_P(\cdot)$. We obtain the similarity score r_i via the inner product of their encoder outputs: $r_i = \text{sim}(x, e_i) = E_Q(x)^T E_P(e_i)$. (2) **Similarity Matrix Construction.** Using similarity scores $\boldsymbol{r} = [r_1, r_2, ..., r_M]$ for all M examples in the example pool D, we construct a diagonal similarity matrix $\boldsymbol{R} \in \mathbb{R}^{M \times M}$: $\boldsymbol{R} = Diag(\boldsymbol{r})$, where $Diag(\cdot)$ is the diagonal matrix operator. The diagonal of $\boldsymbol{R}$ consists of $\boldsymbol{r}$, while all off-diagonal elements are 0. (3) **Kernel Matrix Adaptation.** We adapt the optimized kernel matrix to the given input query by incorporating the above similarity matrix $\boldsymbol{R}$ with the optimized kernel matrix $\hat{\boldsymbol{L}}$ obtained in 4.3. That is, we obtain the adapted kernel matrix $\boldsymbol{L}'$ as: $\boldsymbol{L}' = \boldsymbol{R} \cdot \hat{\boldsymbol{L}} \cdot \boldsymbol{R}$.

4.4.2 Query-Oriented Example Selection via MAP Inference

To select the ICL example set for the query with the adapted NDPP, rather than selecting the most relevant k examples (Rubin et al., 2022; Wang et al., 2024b), we conduct the MAP inference, the standard subset sampling method for NDPP, to select examples one by one from the example pool via greedy algorithm. The goal of MAP inference is to select the high-quality ICL example set S_{map} of size k from the example pool D for the current query. In the adapted NDPP, given the kernel matrix $\boldsymbol{L}'$, S_{map} is the example subset of size k from the example pool D that maximizes $P_{\boldsymbol{L}'}(S)$ among all possible subsets S of size k. Recall that the probability $P_{\boldsymbol{L}'}(S)$ is proportional to the determinant of the sub-kernel matrix $\boldsymbol{L}'_S$, S_{map} is the example subset from the example pool D that maximizes $det(\boldsymbol{L}'_S)$ among all possible subsets S of size k. Formally, we define the MAP inference of the example selection with adapted NDPP as:

$$S_{map} = \underset{S \subseteq D, |S|=k}{\arg\max} \ logdet(\boldsymbol{L}'_S) \tag{10}$$

However, the MAP inference above has been proved to be NP-hard[4] (Ko et al., 1995; Kulesza and Taskar, 2012). To reduce the computational cost, a common approach is to approximate the MAP inference using greedy algorithms (Nemhauser et al., 1978; Gillenwater et al., 2012; Chen et al., 2018). To reduce the cost, we first select a candidate example set $Z, |Z| = K, K < M$ with KNN retriever to reduce the size of candidate examples. Then, following Gartrell et al. (2021), we approximate MAP inference using the greedy algorithm: starting from an empty set S_{map}, we iteratively select examples one by one until we obtained k examples, approximating the global optimum by solving local optima at each iteration. At each iteration, for all examples i in the candidate example set Z that are not included in S_{map}, we compute the increment of the log-determinant $logdet(\cdot)$ of the sub-kernel matrix $\boldsymbol{L}'_{S_{map}}$ after adding example i to the set S_{map}. We select the example j with the largest increment as the local optima and add it into S_{map}:

$$\begin{aligned} j = \underset{i \in Z \setminus S_{map}}{\arg\max} \ & logdet\left(\boldsymbol{L}'_{S_{map} \cup \{i\}}\right) \\ & - logdet\left(\boldsymbol{L}'_{S_{map}}\right) \end{aligned} \tag{11}$$

Finally, we concatenate the query and the ICL example set S_{map} as the input prompt of LLMs.

5 Experiments

5.1 Experiments Settings

Dataset. Following (Ye et al., 2023b; Li et al., 2023), we use five datasets: (1) GeoQuery (Shaw et al., 2020) has 880 geography questions. (2) NL2Bash (Lin et al., 2018) contains 9k Bash command pairs. (3) MTOP (Li et al., 2020) is a multilingual parsing dataset with 6 languages. (4) WebQs (Berant et al., 2013) covers 6,642 QA pairs using Freebase. (5) Roc Ending (Mostafazadeh et al., 2016) is a corpus with 100k stories.

[4] Such MAP inference requires finding all subsets $S, |S| = k$ of the example pool $D, |D| = M$ and computing their determinants. The example pool $D, |D| = M$ has $C(M, k)$ subsets $S, |S| = k$ in total, and the computational complexity of each subset determinant is $O(k^3)$. The cost of the MAP inference is $O(M^k \cdot k^3)$ in total, which is unaffordable as the size of the example pool D increases. And the function $logdet(\boldsymbol{L}'_S)$ is proved to be submodular, and the unconstrained optimization problem for submodular is NP-hard.

Model	Method	GeoQuery (EM)	MTOP (EM)	NL2Bash (BLEU-4)	WebQs (EM)	Roc Ending (BLEU-1)
GPT-Neo (2.7B)	Random	33.57	0.67	34.35	4.87	57.58
	BM25	62.86	53.24	58.98	16.68	58.65
	EPR	71.07	60.36	56.82	17.91	59.12
	CEIL*	70.71	63.4	53.66	17.08	59.72
	TTF*	68.93	54.05	56.11	16.14	/
	Our	**73.21**	**65.37**	**61.01**	**18.9**	**60.33**
GPT-4	Random	71.43	21.48	67.45	34.49	58.34
	EPR	88.93	78.61	73.63	50.32	54.7
	CEIL	91.07	78.7	73.95	46.75	56.24
	Our	**91.43**	**79.02**	**73.96**	**52.95**	**62.81**

Table 1: ICL example selection experiment results. "/" indicates that the method is not open source and does not give results of the dataset in the corresponding paper and "Bold" indicates optimal results. All results are averaged over 3 runs.We reference results from the previous work (Liu et al., 2024b), marked by *. Our improvements are significant under the t-test with p < 0.05 (See details in Appendix B).

Metrics. Following (Ye et al., 2023b; Li et al., 2023), we use those metrics: (1) Exact Match (EM) (Rajpurkar et al., 2016) for GeoQuery, MTOP, and WebQs to assess the accuracy of the generated output. (2) BLEU-1 (Papineni et al., 2002) for Roc Ending to evaluate alignment in story generation. (3) BLEU-4 (Papineni et al., 2002) for NL2Bash to capture longer sequence structure in command generation.

Baselines. We compare with two types of methods: (1) **Unsupervised Methods:** Random, which randomly selects non-repeating ICL examples from the example pool. BM25 (Robertson et al., 2009), which extends TF-IDF to rank relevant examples for the test input and select the top-k highest scoring ICL examples for each test input. (2) **Supervised Methods:** EPR (Rubin et al., 2022), which uses the LLM itself as a scoring model to retrieve good ICL examples. CEIL (Ye et al., 2023b) models ICL example sets with DPP and trains DPP by contrastive learning. TTF (Liu et al., 2024b) fine-tunes the ICL example selector with labeled data, adding task-specific modules.

See the implementation details in Appendix A.

5.2 Overall Performance

Table 1 shows the overall results of ICL example selection methods across five datasets. Notably, while prior studies (Rubin et al., 2022; Ye et al., 2023a) primarily focus on smaller models like GPT-Neo (2.7B), we extend the evaluation to the SOTA LLM GPT-4[5]. The results demonstrate that our method outperforms all baseline methods on both GPT-neo-2.7B and GPT-4 models.

Compared to random selection, our method shows over 20% average improvement on both models. All designed selection methods outperform random selection except for GPT-4 on the Roc Ending dataset, highlighting the value of careful example selection. We observe that the performance improvement of our method is more pronounced on GPT-neo-2.7B compared to GPT-4, likely due to the latter's inherently stronger inference capability. This finding is consistent with previous research (Zhang et al., 2022). However, on the Geoquery, Mtop, and Roc Ending datasets, our method on GPT-neo-2.7B outperforms random example selection on GPT-4, demonstrating the effectiveness of our approach in enhancing the ICL capability of LLMs. Furthermore, our method consistently outperforms CEIL on all datasets, suggesting the benefits of capturing positive correlations among examples for ICL example selection.

5.3 Ablation Study

Table 2 presents the ablation study conducted on our model. Our complete model performs excellently across all five datasets, and removing any single module leads to a decrease in performance, validating the effectiveness of each component. Specifically: (1) w/o Scoring: We remove the step of scoring with LLM and instead use all the example subsets as the training set. We observe that although performance slightly declined, our model still maintains relatively good performance on some tasks. This suggests that our model is still able to model correlations among examples to some extent, but is disturbed by noise in low-scoring ICL example subsets. (2) w/o Regularization: We removed the

[5]Due to the limitations of black-box models like GPT-4 (which only expose log probabilities for the first five tokens), our framework cannot directly construct pseudo-labeled training sets based on full token probabilities. To address this, we transfer the retriever trained on GPT-Neo-2.7B directly to GPT-4 for ICL example selection

Settings	GeoQuery (EM)	MTOP (EM)	NL2Bash (BLEU-4)	WebQs (EM)	Roc Ending (BLEU-1)
Ours(Full Model)	**73.21**	**65.37**	**61.01**	**18.9**	**60.33**
w/o Scoring	72.36	65.19	59.32	17.91	59.09
w/o Regularization	71.43	65.28	60.25	18.75	59.94
w/o Adaptation	71.64	65.28	59.56	18.45	60.33

Table 2: Ablation study. w/o Scoring: remove the LLM scoring when construct the training set; w/o Regularization: remove the regularization term in the log-likelihood; w/o Adaptation: remove query-aware kernel adaptation on the trained NDPP.

	GeoQuery	MTOP	WebQs	Roc Ending
Best Random-Order	69.29	62.64	14.86	59.50
Worst Random-Order	66.43	61.48	13.24	58.10
VAR	0.78	0.13	0.21	0.19

Table 3: The effect of different example orders.

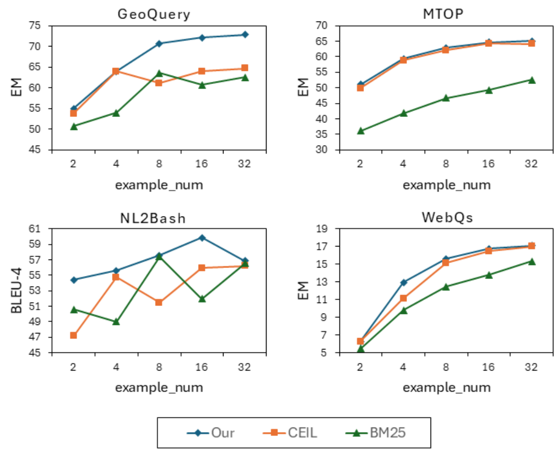

Figure 2: The effect of different example numbers.

regularization term in Eq. 9, and the performance of our model deteriorates on certain tasks. Without regularization, our model exhibits a tendency to overfit, which results in a decrease in generalization ability on test data. (3) w/o Adaptation: We remove the query-aware kernel adaptation and observe a performance drop, which demonstrates the importance of considering the relevance between queries and examples.

5.4 Analysis Study of ICL Example Order

Previous work (Lu et al., 2022) showed that ICL is sensitive to the order of examples when using Randomly selected examples. We conduct experiments to investigate the effect of ordering on ICL examples retrieved by our method. Specifically, we provide 8 examples with 10 different random orderings for each dataset. We present the best (Best Random-Order) and worst (Worst Random-Order) results and the variance of the results over 10 runs. The results are shown in Table 3.

We find that performance fluctuates somewhat across different random orderings, but the variation is relatively small and within a controllable range. This suggests that although example order does have some impact on the performance of our model, the effect is limited. This finding is consistent with previous research (Li and Qiu, 2023), which indicates that high-quality examples can reduce ICL sensitivity to the order of examples.

5.5 Analysis Study of ICL Example Numbers

Many LLMs are constrained by limited input lengths, which restricts the maximum number of in-context learning (ICL) examples that can be provided. To analyze the impact of example quantity on ICL performance, we compared three methods across four tasks, and the results are shown in figure 2. Our key observations are as follows: (1) Increasing the number of examples enhances ICL performance, as additional examples enable LLMs to better understand the task objectives and output patterns. (2) Beyond a certain point (e.g., 16 or 32), the performance gains plateau. This is because the marginal utility of additional examples diminishes, as LLMs' capacity to extract useful information from further demonstrations becomes saturated.

6 Conclusion

In summary, we proposed an NDPP-based framework for ICL example selection. Our framework first constructs a pseudo-labeled training set based on LLM feedback, and then uses the set to train the NDPP model by kernel decomposition-based MLE. Finally, in the inference stage, we perform query adaptation on the NDPP model, followed by MAP inference to select suitable and customized ICL example sets for different queries. Our experiments on five datasets across four domains show that our framework achieves SOTA performance in ICL example selection.

Limitations

The pseudo-labeled training dataset we construct relies on LLM feedback, which may be subject to inherent biases within the LLM. To address this limitation, future work could explore integrating fairness-aware mechanisms into the LLM feedback process, such as debiasing techniques, fairness constraints, or adversarial training, to mitigate potential biases.

Our framework constructs pseudo-labeled datasets based on token probabilities from LLM feedback, which inherently limits its compatibility with black-box models (e.g., GPT-4), as they only expose log probabilities for the top five tokens. However, our experiments demonstrate that a retriever trained on white-box models (e.g., GPT-Neo) can be effectively transferred to black-box models, achieving competitive performance. In future work, we plan to explore alternative approaches for constructing pseudo-labeled datasets that are universally applicable, including black-box LLMs.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873.

Shengnan An, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Jian-Guang Lou, and Dongmei Zhang. 2023. How do in-context examples affect compositional generalization? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11027–11052.

Maurice Stevenson Bartlett. 1937. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 160(901):268–282.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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

Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31.

Xiwen Chen, Huayu Li, Rahul Amin, and Abolfazl Razi. 2023. Rd-dpp: Rate-distortion theory meets determinantal point process to diversify learning data samples. *arXiv preprint arXiv:2304.04137*.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. Uprise: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337.

Wei Duan, Junyu Xuan, Maoying Qiao, and Jie Lu. 2022. Learning from the dark: boosting graph convolutional neural networks with diverse negative samples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6550–6558.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Mike Gartrell, Victor-Emmanuel Brunel, Elvis Dohmatob, and Syrine Krichene. 2019. Learning nonsymmetric determinantal point processes. *Advances in Neural Information Processing Systems*, 32.

Mike Gartrell, Insu Han, Elvis Dohmatob, Jennifer Gillenwater, and Victor-Emmanuel Brunel. 2021. Scalable learning and {map} inference for nonsymmetric determinantal point processes. In *International Conference on Learning Representations*.

Lorenzo Ghilotti, Mario Beraha, and Alessandra Guglielmi. 2024. Bayesian clustering of high-dimensional data via latent repulsive mixtures. *Biometrika*, page asae059.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. 2012. Near-optimal map inference for determinantal point processes. *Advances in Neural Information Processing Systems*, 25.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148.

Julia Grosse, Rahel Fischer, Roman Garnett, and Philipp Hennig. 2024. A greedy approximation for k-determinantal point processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3052–3060. PMLR.

Insu Han, Mike Gartrell, Jennifer Gillenwater, Elvis Dohmatob, and Amin Karbasi. 2022. Scalable sampling for nonsymmetric determinantal point processes. *arXiv preprint arXiv:2201.08417*.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Simon Johansson, Ola Engkvist, Morteza Haghir Chehreghani, and Alexander Schliep. 2023. Diverse data expansion with semi-supervised k-determinantal point processes. In *2023 IEEE International Conference on Big Data (BigData)*, pages 5260–5265. IEEE.

Chun-Wa Ko, Jon Lee, and Maurice Queyranne. 1995. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D Ernst. 2018. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. *arXiv preprint arXiv:1802.08979*.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. se2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5262–5284.

Hui Liu, Wenya Wang, Hao Sun, Chris Xing Tian, Chenqi Kong, Xin Dong, and Haoliang Li. 2024b. Unraveling the mechanics of learning-based demonstration selection for in-context learning. *arXiv preprint arXiv:2406.11890*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Yuli Liu, Christian Walder, and Lexing Xie. 2024c. Learning k-determinantal point processes for personalized ranking. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1036–1049. IEEE.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14:265–294.

Michael Aggrey Okoth, Ronghua Shang, Licheng Jiao, Jehangir Arshad, Ateeq Ur Rehman, and Habib Hamam. 2022. A large scale evolutionary algorithm based on determinantal point processes for large scale multi-objective optimization problems. *Electronics*, 11(20):3317.

Shilong Pan, Zhiliang Tian, Liang Ding, Haoqi Zheng, Zhen Huang, Zhihua Wen, and Dongsheng Li. 2024. POMP: Probability-driven meta-graph prompter for LLMs in low-resource unsupervised neural machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9976–9992, Bangkok, Thailand. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2020. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? *arXiv preprint arXiv:2010.12725*.

Hassam Sheikh, Kizza Frisbee, and Mariano Phielipp. 2022. Dns: Determinantal point process based neural network sampler for ensemble reinforcement learning. In *International Conference on Machine Learning*, pages 19731–19746. PMLR.

Jianbin Shen, Junyu Xuan, and Christy Liang. 2023. A determinantal point process based novel sampling method of abstractive text summarization. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Zhao Song, Junze Yin, Lichen Zhang, and Ruizhe Zhang. 2024. Fast dynamic sampling for determinantal point processes. In *International Conference on Artificial Intelligence and Statistics*, pages 244–252. PMLR.

Huazheng Wang, Jinming Wu, Haifeng Sun, Zixuan Xia, Daixuan Cheng, Jingyu Wang, Qi Qi, and Jianxin Liao. 2024a. Mdr: Model-specific demonstration retrieval at inference time for in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4189–4204.

Liang Wang, Nan Yang, and Furu Wei. 2024b. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering. In *Advances in Neural Information Processing Systems*, volume 37, pages 88906–88931. Curran Associates, Inc.

Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng YANG, Qingxing Cao, Haiming Wang, Xiongwei Han, Jing Tang, Chengming Li, and Xiaodan Liang. 2024. DQ-lore: Dual queries with low rank approximation re-ranking for in-context learning. In *The Twelfth International Conference on Learning Representations*.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023a. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023b. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

A Implementation Details.

We used GPT-neo-2.7B and GPT-4 as LLM for our study. The maximum context length for the input of the LLM was set at 2048 tokens and the number of context examples per task was set to 50. If the context size limit of the LLM is exceeded, it will be truncated. We adopted the Adam optimizer with a learning rate of 0.01, and the hyperparameters α and β were both set to 0.01. The training was conducted on two NVIDIA A100 GPUs. We initialize the encoder $E_Q(\cdot)$ and $E_Q(\cdot)$ with CEIL (Ye et al., 2023a). We employ the implementation from Ye et al. (2023a) for random, BM25, and EPR. For CEIL, we use the result from Liu et al. (2024b) except the result of Roc Ending. We also employ the implementation from Ye et al. (2023a) to obtain the result of Roc Ending for CEIL.

B Significance Test.

We conduct the t-test (Bartlett, 1937) to examine whether the improvements of our method are significant. The p values in Table 4 are all smaller than 0.05, demonstrating the significance of our improvements.

Dataset	Dataset	GeoQuery	NL2Bash	MTOP	WebQs	Roc Ending
GPT-Neo (2.7B)	Bartlett's Test	0	5.73e-61	0	7.29e-05	0
GPT-4	Bartlett's Test	6.92e-03	0.0052	4.01e-12	0.0116	0.0251

Table 4: The p values of t-test on our method with baselines. The p values are all smaller than 0.05, indicating our improvements are significant.