# Prompt Optimization with EASE? Efficient Ordering-aware Automated Selection of Exemplars

**Zhaoxuan Wu***[§¶], **Xiaoqiang Lin***[†], **Zhongxiang Dai**[ς], **Wenyang Hu**[§†], **Yao Shu**[‖], **See-Kiong Ng**[§†], **Patrick Jaillet**[‡], **Bryan Kian Hsiang Low**[†]
Institute of Data Science, National University of Singapore, Republic of Singapore[§]
Singapore-MIT Alliance for Research and Technology, Republic of Singapore[¶]
Dept. of Computer Science, National University of Singapore, Republic of Singapore[†]
School of Data Science, The Chinese University of Hong Kong, Shenzhen[ς]
LIDS and EECS, Massachusetts Institute of Technology, USA[‡]
Guangdong Lab of AI and Digital Economy (SZ)[‖]
{wu.zhaoxuan, xiaoqiang.lin, wenyang.hu}@u.nus.edu[§]
daizhongxiang@cuhk.edu.cn[ς], seekiong@nus.edu.sg[§]
lowkh@comp.nus.edu.sg[†], jaillet@mit.edu[‡], shuyao@gml.ac.cn[‖]

## Abstract

Large language models (LLMs) have shown impressive capabilities in real-world applications. The capability of *in-context learning* (ICL) allows us to adapt an LLM to downstream tasks by including input-label exemplars in the prompt without model fine-tuning. However, the quality of these exemplars in the prompt greatly impacts performance, highlighting the need for an effective automated exemplar selection method. Recent studies have explored retrieval-based approaches to select exemplars tailored to individual test queries, which can be undesirable due to extra test-time computation and an increased risk of data exposure. Moreover, existing methods fail to adequately account for the impact of exemplar ordering on the performance. On the other hand, the impact of the *instruction*, another essential component in the prompt given to the LLM, is often overlooked in existing exemplar selection methods. To address these challenges, we propose a novel method named `EASE`, which leverages the hidden embedding from a pre-trained language model to represent ordered sets of exemplars and uses a neural bandit algorithm to optimize the sets of exemplars *while accounting for exemplar ordering*. Our `EASE` can efficiently find an ordered set of exemplars that *performs well for all test queries* from a given task, thereby eliminating test-time computation. Importantly, `EASE` can be readily extended to *jointly optimize both the exemplars and the instruction*. Through extensive empirical evaluations (including novel tasks), we demonstrate the superiority of `EASE` over existing methods, and reveal practical insights about the impact of exemplar selection on ICL performance, which may be of independent interest. Our code is available at `https://github.com/ZhaoxuanWu/EASE-Prompt-Optimization`.

## 1 Introduction

Large language models (LLMs) have recently drawn significant attention and have been widely deployed in various real-world scenarios [26, 28, 51] due to their strong capabilities. Of note, a particularly impressive capability of LLMs is *in-context learning* (ICL): LLMs can learn from a

* Equal contribution. Correspondence to: Zhongxiang Dai <daizhongxiang@cuhk.edu.cn>.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3899

handful of input-label demonstrations (i.e., *data exemplars*) included in its prompt to perform a downstream task [3, 23]. ICL allows us to adapt an LLM to a downstream task without fine-tuning the model parameters [20, 29]. However, the ICL performance is heavily dependent on the data exemplars in the prompt [1, 23, 38]. Therefore, to maximize the ICL performance, it is of paramount importance to carefully select the set of exemplars [46]. Unfortunately, exemplar selection for ICL is challenging because the mechanism of ICL is complicated and unknown, and this difficulty is further aggravated for black-box LLMs to which we only have API access (e.g., GPT-4 [32], Gemini Ultra [41]).

A large number of existing works on exemplar selection for ICL have considered *retrieval-based* methods [1, 23, 38]. Specifically, they aim to develop a retriever model such that for every test query, the retriever model retrieves the corresponding best set of exemplars tailored to this particular query [1, 48]. Such approaches require test-time retrieval for every query which might hinder their ease of adoption [25] while test-time computation is not needed for our method. Moreover, another drawback of these retrieval-based methods is that they may lead to increased privacy risks. Compared to using a fixed set of exemplars for all test queries submitted to the LLM provider (e.g., via an API), using a different set of exemplars for every test query may lead to greater data exposure, i.e., more data exemplars being exposed to the LLM provider. Privacy issues have become increasingly prominent in discussions surrounding LLMs [30], with evidence of user data being output by these models [45]. Therefore, retrieval-based methods are undesirable in scenarios where such privacy concerns are important considerations. Given these two important shortcomings of retrieval-based methods, it is imperative to develop exemplar selection methods that generalize to test queries and are capable of choosing a fixed set of exemplars that perform well for all test queries for a task.

Another major limitation of existing exemplar selection methods (including retrieval-based and other methods) is their inability to account for the impact of the *ordering of the exemplars* within a subset [24, 33, 52]. Specifically, to select a subset of $k$ exemplars, previous works have either relied on simple heuristics (e.g., selecting the top-$k$ exemplars based on the score from a retriever [25]) or used subset selection techniques which are able to account for the inter-relationships among the exemplars within a subset [4, 14, 18, 19, 48]. Due to the computational complexity caused by the combinatorial search space, these works based on subset selection have often employed an iterative greedy strategy to sequentially select the subset of exemplars. It is also non-trivial to extend these existing works to consider exemplar ordering. However, it has been repeatedly verified that the ordering of the exemplars has a significant impact on the performance of ICL [24, 33, 52]. To the best of our knowledge, the impact of the exemplar ordering during the selection process remains inadequately addressed by these previous works that relied on simple heuristics (e.g., random ordering) and approximations (e.g., greedy strategy).

In addition, when optimizing the set of exemplars to improve the performance of the LLM, existing methods have often ignored the impact of the *instruction*, which is another essential component in the prompt given to the LLM [21]. Specifically, previous works on exemplar selection often use a fixed pre-determined instruction or do not include any instruction at all in the prompt [4, 19, 23, 38]. Meanwhile, existing works on instruction optimization for LLMs typically include a fixed manually selected set of exemplars in the prompt [10, 16, 21, 47] or simply adopt the zero-shot setting (i.e., without using any exemplar) [5, 13]. However, these two lines of work have separately demonstrated that both the exemplars and the instruction have significant impacts on the performance of the LLM. Therefore, the existing works that optimize these two components separately are unable to account for their interactions, which may be crucial for further boosting the performance of LLMs. This leaves considerable untapped potential in enhancing the performance of the LLM through ICL, which can be achieved by *jointly optimizing the instruction and the exemplars*.

In this work, we propose a novel exemplar selection algorithm that addresses the above-mentioned challenges faced by existing works (with detailed discussions of related work in App. 5). We formulate exemplar selection as a black-box optimization problem, in which every input in the domain corresponds to a sequence of $k$ exemplars and its corresponding output represents the ICL performance achieved by including this sequence in the prompt for the LLM. Given this formulation, for every sequence of $k$ exemplars in the domain, we adopt the embedding from a powerful pre-trained language model as its continuous representation, and train a neural network (NN) to predict its ICL performance. Based on the trained NN, we use a neural bandit algorithm to find the optimal exemplar sequence in a query-efficient manner. Our Efficient ordering-aware Automated Selection of Exemplars (`EASE`) algorithm offers several significant benefits:

- Our EASE can *find an efficacious sequence of $k$ exemplars* (which performs well for all test queries from a task) in a *query-efficient manner* (i.e., it only needs to test a small number of exemplar sequences). This can mainly be attributed to our algorithmic design, which allows us to use neural bandits to balance the *exploration* of the space of exemplar sequences and the *exploitation* of the predicted performance from the NN in a principled way. Importantly, in contrast to retrieval-based methods, our EASE *requires no test-time computation* to select test query-specific exemplars.
- Our EASE naturally *takes into account the ordering of the exemplars* when maximizing the ICL performance. This is because, for the same subset of exemplars, a different ordering leads to a different pre-trained embedding, which allows the trained NN (that takes the embedding as input) to predict the performances of different orderings of these exemplars. We have made our EASE computationally feasible for large search spaces of exemplar sequences using a technique based on optimal transport (OT), which reduces the computational cost of EASE while preserving its strong performance by imposing an implicit preference towards exemplars that are more relevant to the task (Sec. 3.2).
- Our EASE is readily extended to *jointly optimize the exemplars and instruction* in the prompt. This is achieved by augmenting the domain of exemplar sequences with the instruction (Sec. 3.3).

These advantages of our EASE algorithm allow it to significantly boost the performance of ICL. To empirically validate this, we compare our EASE algorithm with a comprehensive suite of baselines in a variety of tasks. To begin with, we show that our EASE consistently outperforms previous baselines in large number of benchmark tasks (Sec. 4.1). Next, by using our EASE algorithm in a novel experiment, we unveil an interesting insight about ICL: The selection of exemplars is more important when the LLM has less knowledge about the task (Sec. 4.2). Based on this insight, we design a set of novel experiments in which the selection of exemplars has an important impact on the ICL performance, and use these experiments to further demonstrate the superiority of our EASE (Sec. 4.3). Furthermore, we showcase the ability of our EASE to jointly optimize the exemplars and the instruction to enhance the performance of the LLM even further (Sec. 4.4). We also included a retrieval-based extension of EASE to deal with large exemplar set sizes (Sec. 4.5). Of note, our novel experimental designs, as well as the insights from them, *may be of independent interest for future works on ICL and beyond.*

## 2 Problem setting

We are given a set of data exemplars $D = \{e_i = (x_i, y_i)\}_{i=1}^n$ of size $n$, where $x_i$ and $y_i$ correspond to the input and output text, respectively. The data exemplars describe a downstream task, and we aim to select $k$ of them to form the optimal in-context exemplars sequence $E$ for accurate output generation when prompting a black-box LLM $f(\cdot)$. Due to the sequential nature of the natural language input, the exemplar sequence is ordered such that $E = (e_1, e_2, \ldots, e_k)$ where $e_i$ denotes the $i$-th exemplar chosen. For every input $x$, we prepend it with a sequence of exemplars $E$ to generate a response $\hat{y}$ from the black-box LLM (e.g., through calling the API), following

$$\hat{y} = f([\underbrace{e_1, e_2, \ldots, e_k}_{\text{context}}, x]) = f([E, x]) \; .$$

Here, the number of exemplar $k$ in the context can be determined by the future budget of inference in practice. A larger $k$ corresponds to a longer sequence of context tokens to be prepended to the test input $x$ during inference, which leads to higher query costs (e.g., associated with the API calls). Alternatively, it is common to decide $k$ depending on the context window size of the target LLM $f(\cdot)$.

Since the model architecture and the internal working of the black-box LLM $f(\cdot)$ is unattainable, we formulate the exemplar selection as a black-box optimization problem over the space of permutations $\Omega \triangleq \{E : |E| = k\}$ of size $n^k$ (i.e., having $n$ choices for each of the $k$ positions in the sequence),

$$\max_{E \in \Omega} F(E) \triangleq \mathbb{E}_{(x,y) \in D_V}[s(f(E, x), y)] \tag{1}$$

where $s(\cdot, \cdot)$ is a score function for the output against the ground truth, $D_V$ is the held-out validation set and $|E| = k$ denotes that the number of exemplars in $E$ is $k$. Therefore, the exemplar sequence $E$ found represents a fixed ordered set of exemplars that apply to every data point in the validation set. Note that our formulation considers exemplars jointly as *ordered* text in the sequence $E$. This space of permutation $\Omega$ is also a lot larger than the (unordered) combinatorial search space considered in the subset selection formulation of exemplar selections in [4, 14, 19, 48]. We show the superior performance of our method against subset selection methods in our experiments (Sec. 4).

 https://doi.org/10.52202/079017-3899

## 3 Automated selection of exemplars

### 3.1 NeuralUCB for query-efficient optimization of exemplar sequences

We propose to use neural bandits to iteratively maximize the black-box objective function (1). At each iteration $t$, the neural bandits algorithm selects the next input query based on the belief of the objective given all past $t-1$ observations $O_{t-1} \triangleq \{(E_i, s_V(E_i))\}_{i=1}^{t-1}$ where $E_i$ and $s_V(E_i)$ are the exemplar sequence and corresponding validation score at iteration $i$, respectively. Here, the validation score is a realization of the objective function (1), so $s_V(E) = 1/|D_V| \sum_{(x,y)\in D_V} s(f(E,x),y)$.

Inspired by Lin et al. [21] who have shown impressive performances of the neural upper confidence bound (NeuralUCB) acquisition function [53] on instruction optimization, we adopt the NeuralUCB approach to our novel black-box exemplar selection formulation in (1). We propose to use a neural network (NN) to directly learn the mapping from a general-purpose hidden embedding of input exemplar sequences to the validation score. Specifically, we propose to use an embedding model $h(\cdot)$ and train a network $m(h(E);\theta_t)$ at iteration $t$ such that

$$\theta_t = \arg\min_\theta \mathbb{E}_{(E,s_V(E))\in O_{t-1}} \ell\left(m(h(E);\theta), s_V(E)\right) \tag{2}$$

where $\ell$ is the mean squared error (MSE) loss function. The embedding model can be pre-trained on a large corpus of text and hence provides a powerful latent representation of the input sentence. For example, pre-trained sentence-transformer models like MPNet [40] and black-box APIs like OpenAI text embedding are both commonly used for downstream clustering, semantic search and classification. Importantly, even for the same subset of exemplars, a different ordering leads to different embedding, hence $h(E)$ captures both the content and the ordering of the exemplars in $E$.

Then, the trained NN can be used to iteratively select the next exemplar sequence $E_t$ to query:

$$\begin{aligned} E_t &= \arg\max_{E\in\Omega} \text{NeuralUCB}_t(E), \\ \text{NeuralUCB}_t(E) &\triangleq m(h(E);\theta_t) + \nu_t \sigma_{t-1}(h(E);\theta_t), \end{aligned} \tag{3}$$

where $m(h(E);\theta_t)$ is the predicted score, $\sigma_{t-1}(h(E);\theta_t)$ is the NN's uncertainty about the score of $h(E)$ and $\nu_t$ is a hyperparameter that balances the two terms. Using NeuralUCB, our `EASE` balances the *exploration* of the space of exemplar sequences $E$ and the *exploitation* of the predicted score from the NN in a principled way.

This application of NeuralUCB in our work differs from that of Lin et al. [21] in three main aspects. Firstly, the input to our NN is the embedding of *exemplar sequences* rather than the latent representation of the *instructions* from [21]. Secondly, we relax the requirement of a separate white-box LLM from [21] and only need black-box access to the embedding model $h(\cdot)$. Thirdly, our application to exemplar selection dramatically increases the search space from finite candidate instructions [21] to the space of exemplar sequences. Particularly, the explosion of the size of the search space poses significant difficulties to optimization, which we will address in the next section.

### 3.2 Reducing computational cost through optimal transport (OT)

Directly applying NeuralUCB to exemplar selection is challenging due to the enormous search space of all permutations of exemplars on which the acquisition values (3) have to be evaluated. For example, selecting 5 exemplars out of 100 requires $100^5$ evaluations which is far from being practical. It is natural to search over a *domain space* $Q_t = \{E^{(j,t)}\}_{j=1}^{q}$ with a smaller number $q$ of exemplar sequences in each iteration $t$. However, uniformly sampling such a reduced space $Q_t$ from $\Omega$ could be sub-optimal, because a small $q$, which is required to make our algorithm computationally feasible, is highly likely to discard important exemplar sequences (we verify this in ablation experiments in Sec. 4.5 and App. C.3). To this end, we make a large $q$ feasible by introducing a novel technique based on optimal transport (OT) to further select from $Q_t$ a subset $Q'_t$ of $q' < q$ *relevant* exemplar sequences. This technique can simultaneously (a) reduce the computational cost (since $q' \ll \Omega$) and (b) preserve our performance (since we can now search over a large domain space $Q_t$) by imposing an implicit preference towards exemplars in $Q_t$ that are more relevant to the task.

Given probability measures $\mu_s$ and $\mu_v$ over space $\mathcal{Z}$, the OT distance between $\mu_s$ and $\mu_v$ is defined as

$$OT(\mu_s,\mu_v) = \min_{\pi\in\Pi(\mu_s,\mu_v)} \int_{\mathcal{Z}^2} c(z,z')d\pi(z,z') \tag{4}$$

where $\Pi(\mu_s, \mu_v) = \{\pi \in \mathcal{P}(\mathcal{Z} \times \mathcal{Z}) | \int_{\mathcal{Z}} \pi(z, z')dz = \mu_s, \int_{\mathcal{Z}} \pi(z, z')dz' = \mu_v\}$ is a collection of couplings between two the distributions $\mu_s$ and $\mu_v$, and $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}^+$ is some symmetric positive cost function. In our problem, we propose to use the space of embedding from $h(\cdot)$ as $\mathcal{Z}$ because the embedding captures the semantics of the exemplars with a fixed-dimensional vector. As cosine similarity is usually used to train embedding models such as MPNet [40] and sentence-BERT [36], we propose to use the following cost function $c(h(e), h(e')) = 1 - sim_{cos}(h(e), h(e'))$ where $sim_{cos}(h(e), h(e'))$ measures the cosine similarity between the embedding of two exemplars $e$ and $e'$. Given a sampled subset $S = \{e_i\}_{i=1}^k$, we define a discrete measure $\mu_s = \frac{1}{k}\sum_{i=1}^k \delta(h(e_i))$ where $\delta$ is the Dirac function. Likewise, we define $\mu_v$ for the validation set $D_V$.

Intuitively, a smaller value of (4) indicates that the subset of exemplars is more similar to the validation set and is hence more *relevant* to the task. This allows us to select the relevant exemplar sequences from $Q_t$ to form the smaller subset $Q'_t$, on which the acquisition values (3) are evaluated. Consequently, we can increase the size $q$ of $Q_t$ by hundreds of folds while being computationally feasible since the most expensive computation (i.e., computing embeddings for all exemplar sequences in $Q_t$) is not needed. Note that the extra computation incurred by OT is minimal since the embedding of every data exemplar in $D$ and $D_V$ can be pre-computed and reused. Therefore, OT helps us examine a large number of permutations among the space of all permutations without significantly increasing the computation, which helps mitigate the problem of the exploding search space.

### 3.3 Natural extension to jointly optimize instructions and exemplars

Our algorithm can further boost the performance of ICL for LLMs by jointly optimizing the exemplars and the instruction. A natural extension of our formulation in Sec. 2 allows the instruction, being another essential component of the LLM prompt, to be simultaneously optimized with exemplars. This ensures the optimal matching between instructions and exemplars to achieve a fully automated pipeline with superior performance. Specifically, our formulation naturally extends to $E = (p, e_1, e_2, \ldots, e_k)$ where $p \in P$ is an instruction from a candidate space/set $P$, i.e., $Q'_t \leftarrow P \times Q'_t$. Subsequently, $p$ can be intuitively treated as another type of "exemplar" from a new space $P$ and optimized in conjunction with the exemplars. In practice, the fixed set $P$ of candidate instructions can be generated by the black-box model through techniques such as APE [54], PromptBreeder [10], etc. This extension is not only simple in its implementation but also proven to be effective in our experiments in Sec. 4.4.

### 3.4 Our EASE algorithm

In iteration $t$ of EASE, we first use the historical observations of the exemplar sequence and score pairs $\{(E_i, s_V(E_i))\}_{i=1}^{t-1}$ to train the score prediction NN. Then, we perform sampling to obtain the domain space $Q_t$, which will be further refined to a set $Q'_t$ of top-$q'$ candidates based on OT distances. The space of exemplar sequence $E$ can be optionally augmented with instructions $p \in P$ from a set $P$ of instruction candidates. Subsequently, we find the optimal $E$ that maximizes the NeuralUCB acquisition function. The selected exemplar sequence $E$ is then evaluated against the black-box LLM $f(\cdot)$ using the validation dataset $D_V$, obtaining a new observed pair of $(E, s_V(E))$. This process is repeated until the query budget $T$ is exhausted. An overview of the algorithm is presented in Alg. 1.

**Algorithm 1:** EASE

**Require :** Data exemplars set $D$, validation set $D_V$, length of exemplars $k$, total budget $T$, number of initial rounds $T_{\text{init}}$, sampling size $q'$, black-box target model $f(\cdot)$, embedding model $h(\cdot)$, neural network $m(\cdot; \theta)$, *(optional)* instruction set $P$.

Initialize $\{(E_i, s_V(E_i))\}_{i=1}^{T_{\text{init}}}$ with $T_{\text{init}}$ randomly sampled $\{E_i\}_{i=1}^{T_{\text{init}}}$ from $D$;

**for** $t = T_{init}$ **to** $T$ **do**

  Following (2), use observations $\{(E_i, s_V(E_i))\}_{i=1}^{t-1}$ to train the NN $m(\cdot; \theta_t)$ parameter $\theta_t$;

  Sample sequences $Q_t$ and select top-$q'$ sequences $Q'_t$ with smallest OT distances;

  *(Optional)* Augment each $E \in Q'_t$ with instructions $p \in P$, i.e., $Q'_t \leftarrow P \times Q'_t$;

  Following (3), select the next query $E_t = \arg\max_{E \in Q'_t} \text{NeuralUCB}_t(E)$;

  Evaluate $E_t$ on the black-box model to obtain the validation score $s_V(E_t)$;

**return** $E^* = \arg\max_{E_t : t \in \{1, \ldots, T\}} s_V(E_t)$


https://doi.org/10.52202/079017-3899

Table 1: Average accuracy ± standard error achieved by the best exemplar sequence discovered by different algorithms over 3 independent trials. For better distinguishability, we do not include easy tasks here (i.e., with 100% accuracy across baselines) and show full results in Tab. 5 of App. C.1.

| | DPP | MMD | OT | Cosine | BM25 | Active | Inf | Evo | Best-of-N | EASE |
|---|---|---|---|---|---|---|---|---|---|---|
| antonyms | 70.0±0.0 | 80.0±0.0 | 81.7±1.7 | 85.0±0.0 | 85.0±0.0 | 80.0±0.0 | 86.7±1.7 | 88.3±1.7 | **90.0±0.0** | **90.0±0.0** |
| auto_categorization | 3.3±1.7 | 8.3±1.7 | 0.0±0.0 | 25.0±0.0 | 16.7±1.7 | 10.0±2.4 | 21.7±1.7 | 21.7±1.7 | 20.0±0.0 | **30.0±0.0** |
| diff | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| larger_animal | 70.0±0.0 | 91.7±1.7 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** | 66.7±1.4 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| negation | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** | **95.0±0.0** |
| object_counting | 55.0±2.9 | 56.7±1.7 | 48.3±1.7 | 61.7±1.7 | 66.7±1.7 | 51.7±1.4 | 63.3±4.4 | 70.0±0.0 | 70.0±0.0 | **73.3±1.7** |
| orthography_starts_with | 20.0±2.9 | 35.0±0.0 | 61.7±1.7 | 78.3±1.7 | 70.0±0.0 | 43.3±1.4 | 70.0±2.9 | 75.0±0.0 | 78.3±1.7 | **80.0±0.0** |
| rhymes | 60.0±0.0 | 51.7±1.7 | 0.0±0.0 | **100.0±0.0** | 80.0±0.0 | 65.0±8.2 | 70.0±13.2 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| second_word_letter | 10.0±2.9 | 30.0±0.0 | 28.3±1.7 | 50.0±0.0 | 50.0±0.0 | 26.7±8.3 | 40.0±0.0 | 46.7±1.7 | 50.0±0.0 | **53.3±1.7** |
| sentence_similarity | 20.0±0.0 | 21.7±3.3 | 40.0±2.9 | 46.7±1.7 | 53.3±1.7 | 5.0±4.1 | 18.3±6.7 | 45.0±0.0 | 51.7±1.7 | **56.7±1.7** |
| sentiment | 85.0±0.0 | 90.0±0.0 | 85.0±0.0 | 96.7±1.7 | **100.0±0.0** | 85.0±4.1 | 91.7±1.7 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| sum | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | 0.0±0.0 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| synonyms | 10.0±0.0 | 25.0±0.0 | 20.0±0.0 | **35.0±0.0** | 30.0±0.0 | 3.3±1.4 | 26.7±1.7 | 30.0±0.0 | 30.0±0.0 | 30.0±0.0 |
| taxonomy_animal | 43.3±4.4 | 40.0±2.9 | 46.7±1.7 | 85.0±2.9 | 80.0±0.0 | 45.0±6.2 | 70.0±5.0 | 80.0±0.0 | 80.0±0.0 | **88.3±1.7** |
| translation_en-de | **90.0±0.0** | 80.0±0.0 | 80.0±0.0 | **90.0±0.0** | 85.0±0.0 | 56.7±13.0 | **90.0±0.0** | **90.0±0.0** | **90.0±0.0** | **90.0±0.0** |
| translation_en-es | 90.0±0.0 | **100.0±0.0** | 96.7±1.7 | **100.0±0.0** | **100.0±0.0** | 96.7±1.4 | 98.3±1.7 | **100.0±0.0** | **100.0±0.0** | **100.0±0.0** |
| translation_en-fr | 76.7±1.7 | 76.7±1.7 | 81.7±1.7 | 85.0±0.0 | 85.0±0.0 | 81.7±1.4 | 85.0±0.0 | 86.7±1.7 | 85.0±0.0 | **88.3±1.7** |
| word_sorting | 26.7±1.7 | 88.3±1.7 | 88.3±1.7 | 90.0±0.0 | 71.7±1.7 | 80.0±0.0 | 88.3±1.7 | **93.3±1.7** | 91.7±1.7 | 90.0±0.0 |
| word_unscrambling | 68.3±1.7 | 56.7±1.7 | 71.7±1.7 | 75.0±0.0 | 76.7±1.7 | 63.3±3.6 | 66.7±1.7 | 75.0±0.0 | 75.0±0.0 | **78.3±1.7** |
| # best-performing tasks | 2 | 2 | 2 | 6 | 4 | 1 | 5 | 9 | 9 | **17** |

# 4 Experiments

We compare EASE with the following comprehensive suite of baselines. They include *subset selection methods* with (a) a determinantal point process (**DPP**) metric adapted from [48], and (b) maximum mean discrepancy (**MMD**) [39], (c) optimal transport (**OT**) [42] metrics. We also adapt retrieval-based methods to our setting using a new retrieve-then-sample strategy based on the validation set (details in App. B.2 and Sec. 4.5), specifically with the classical (d) **Cosine** similarity and (e) **BM25** [37] retrievers. We also compare with existing exemplar selection baselines using (f) an active selection policy learned using reinforcement learning (**Active**) [50], and (g) an exemplar influence metric (**Inf**) [31]. Additionally, we propose two more new strong baselines: (h) **Evo** which mutates exemplars through evolutionary strategies and (i) **Best-of-N** which explores the exemplar space uniformly until the query budget is exhausted. Evo is similar to PhaseEvo proposed by Cui et al. [8]. Best-of-N shares similarity with DSPy's BootstrapFewShot with RandomSearch [17]. More implementation details are found in App. B.2. The number of exemplars in the in-context prompt is set to $k = 5$. The black-box query budget is 165 evaluations following Lin et al. [21]. Since the effectiveness of the optimization is directly reflected by the value of the objective function in (1), we report the validation accuracy in the following experiments unless otherwise specified. The test accuracy tables are presented in App. C.15.

## 4.1 Empirical study of performance gains on various benchmark tasks

To study the effectiveness of EASE, we conduct empirical comparisons with baseline methods using the Instruction Induction (II) benchmark tasks [5]. We test on tasks that contain more than 100 training examples for selection. The results are shown in Tab. 1. Our EASE achieves the best performance in 17 out of 19 language tasks. We observe that the subset selection methods such as DPP, MMD and OT are ineffective in practice, implying that choosing subsets alone without considering the order information is inadequate. While the retrieval-based methods Cosine and BM25 have demonstrated success in retrieving test-sample-based exemplars [38], it is not as effective in finding the best exemplars that generalize to the entire task. This is because the exemplars retrieved at the instance level may not work well for the entire validation set. The Active baseline requires training an active exemplar selection policy through the data-intensive reinforcement learning process, which explains its ineffectiveness under our budget-constrained setting. The Inf baseline is not efficient in subset sampling and disregards exemplar ordering, leading to worse results. Our proposed Evo and Best-of-N are the most competitive baselines both with best performance in 9 tasks.

We discover that while EASE consistently outperforms or matches the baselines across tasks, for some tasks, most baselines achieve good performances. We hypothesize that *the effect of exemplar selection diminishes as the model gains more knowledge about the task*. This explains the instances where the choice of in-context exemplars has minimal impact on the performance, as the black-box model (i.e., GPT-3.5 Turbo) has been pre-trained on these publicly available language tasks. As

suggested by Brown et al. [3], it is highly likely that GPT-3 has been trained on common benchmark datasets potentially contained in Common Crawl due to data contamination. Hence, further providing high-quality in-context exemplars could hardly improve the performance of the model on these well-trained tasks. Another possibility could be that the exemplars mostly serve as the context for adapting to the new formatting rules of the specific task, i.e., the LLM is not utilizing the semantic contents of the exemplars to learn the underlying input-output relations. This aligns with the discoveries of Min et al. [27] and provides a potential explanation for the surprising behavior of LLMs. Therefore, the II dataset might not be the most suitable one to test EASE. Next, we verify the above hypothesis that the effect of exemplar selection diminishes as the model gains more knowledge about the task in Sec. 4.2, and propose more suitable families of datasets for exemplar selection in Sec. 4.3.

### 4.2 Empirical validation of the hypothesis through progressive finetuning

We hypothesize that *the effect of exemplar selection diminishes as the model gains more knowledge about the task*. To gain insights on this hypothesis, we study an open-source white-box Vicuna model instead of the black-box GPT-3.5 model that is only accessible through API. We progressively finetune the while-box language model, and examine whether the extent of finetuning on a specific task diminishes the importance of in-context exemplars when prompted.

Our results are in Fig. 1. Compared to the most competitive baseline, Best-of-N, *the gain from exemplar selection using our EASE diminishes as the model is finetuned on the dataset of the respective tasks for more epochs*. Across the three tasks shown in Fig. 1, using EASE originally has a performance gain of about 3%-10% and this gain slowly diminishes to 0 as finetuning progresses. This verifies our hypothesis and calls for further investigation of the exemplar selection performance of our EASE on tasks that have not been seen by the model, which we conduct in Sec. 4.3.

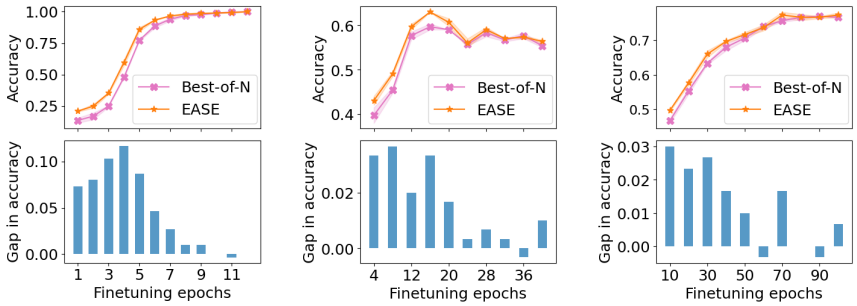


Figure 1: From left to right, the tasks are taxonomy animal, sentence similarity and object counting. The performance gaps between EASE and the Best-of-N baseline diminish as the LLM is finetuned.

### 4.3 New families of "out-of-distribution" tasks that emphasize in-context reasoning

We propose three new families of "out-of-distribution" tasks (i.e., loosely referred to as tasks on which the LLM is not already well trained) that highlight the importance of high-quality exemplars, which could also be of independent interest. The new tasks require the LLM to learn the underlying function/relationship in the input prompt in order to perform reasoning during inference, and are hence more sensitive to the quality of the exemplars.

**Rule-based tasks.** Given our insight on the impact of the model's existing knowledge about the task (Sec. 4.2), we propose rule-based tasks that contain novel rules that the LLM has not learned before. A key characteristic of these tasks is that the model has to extract the underlying relationships among the provided in-context exemplars and directly use the relationship for test-time inferences. For example, we construct the linear regression (**LR**) task where the input takes the form demonstrated in Example 2 (see App. B.1). The underlying relationship in this example is $y = ax + b$, with $a = -4$ and $b = 6$ in this specific Example 2. Without further instructions, the model is supposed to rely on the provided in-context exemplars to implicitly infer the regression task, recover the coefficients $a, b$ for linear regression, and then directly apply it to the test sample (e.g., for test input 117, compute $-4 \times 117 + 6 = -462$ and output $-462$). The results are in Tab. 2. For the clean dataset (i.e., 0% noise), EASE outperforms the most competitive baseline by 8.3% in absolute accuracy. Note that EASE has greater advantages in settings with noisy data, which will be discussed later in this section.


https://doi.org/10.52202/079017-3899

Another example of our proposed rule-based tasks is constructed by changing the rules for language puzzles (**LP**). A prominent example is "pig Latin" which follows two simple rules: (a) For words that begin with a vowel, one just adds "yay" to the end; (b) for words that begin with consonants, the initial consonants are moved to the end of the word, then "ay" is added. For example, translating "Hello, how are you today?" to pig Latin gives "Ellohay, owhay areyay ouyay odaytay?". Note that this task also requires the model to reason about the language translation rules (i.e., rules (a) and (b)) before predicting for the test sample. The rules can be freely modified (e.g., by changing the suffix word "yay" to others) to create diverse tasks that require in-context reasoning from the LLM. Hence, we create a new variant of the LP task, named **LP-variant**, by using the "ay" suffix for both rules (a) and (b). An example of the task query is given in Example 3 (see App. B.1). As shown in Tab. 2, in this task, our EASE also demonstrates competitive results and outperforms all baselines.

**Remapped label tasks.** By remapping the labels of existing classification tasks to new ones, we construct novel tasks that are against the model's existing knowledge. In the commonly used AG News dataset, the news articles are classified into four categories: "World", "Sports", "Business" and "Sci/Tech". We construct a remapped dataset **AG News Remap** such that "World" news is now labeled as "Sports" news, "Sports" is now labeled as "Business", etc. This is against the LLM's knowledge since the output now does not correspond to the context descriptions of the input (i.e., news articles). So, the LLM has to learn these remapping rules from the in-context exemplars. Similarly, we construct another dataset **SST5 Reversed** by reversing the labels of the sentiment analysis dataset SST5, such that "very negative" labels are swapped with "very positive" labels, and "negative" labels are swapped with "positive" labels. The results for these novel remapped label tasks are also in Tab. 2, which shows superior performances of EASE over baselines by 6.7%-10% in absolute accuracy.

**Noisy tasks.** Exemplar selection is even more important to achieve good ICL performances when the dataset is potentially noisy, since using noisy or mislabeled data as exemplars could have detrimental effects on performance. Noisy datasets also better resemble practical scenarios because clean data is expensive and difficult to obtain. We construct noisy datasets by injecting various ratios of noisy outputs into the task datasets, ranging from having 10% noisy data samples (i.e., the remaining 90% are clean) to having 90% noisy data samples (i.e., the remaining 10% are clean). We show in Tab. 2 that EASE is the best-performing method across different noise ratios and generally exhibits a lower decrease in accuracy as the tasks become more difficult with increasing noise ratios.

Note that the conclusions on test accuracy results (in App. C.15) are consistent with the main text. These tasks above represent new families of tasks that contain novel knowledge for LLMs. We show through comprehensive experiments that exemplar selection is important and useful in practice, especially for novel downstream task adaptations. Also, the noisy datasets considered here align with real-world scenarios, which typically contain noises from observation, labeling error, corruption, etc.

### 4.4 Jointly optimizing instruction and exemplars

To the best of our knowledge, no prior work can jointly optimize instruction and exemplars. For a fair comparison, we do not include an instruction when comparing with other exemplar selection baselines in earlier sections. As EASE is readily extended to find the optimal combination of instruction and exemplars (Sec. 3.3), we show the benefits of joint optimization in this section. According to Tab. 3, jointly optimizing these two essential components of a prompt significantly improves the performance for most tasks (marked with red arrows ↑). This improvement is attributed to the ability of the optimized instruction to significantly reinforce the information captured in the corresponding exemplars. For example, an automatically optimized instruction "identify and list the animals from the given words" complements the exemplars in the taxonomy animal task. Therefore, our joint optimization of instruction and exemplars presents a *fully automated pipeline* for prompt optimization and achieves impressive practical performances.

### 4.5 Ablation studies

**Combination with retrieval-based methods to handle larger sets of exemplars.** Applying EASE to scenarios where the size $n$ of the set of exemplars $D$ is very large may lead to performance degradation. This is because the space $\Omega$ of exemplar sequences becomes excessively large, which cannot be sufficiently explored without significantly increasing the computation (i.e., with the size of $Q_t$ being fixed). To resolve this issue, a natural idea is to first filter the large pool of data exemplars to eliminate

Table 2: Average accuracy ± standard error over 3 independent trials achieved by different algorithms on the new families of out-of-distribution tasks.

| Type | Task | Noise | DPP | MMD | OT | Cosine | BM25 | Active | Inf | Evo | Best-of-N | EASE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule-based tasks | LR | 0% | 31.7±1.7 | 38.3±3.3 | 50.0±0.0 | 71.7±1.7 | 70.0±0.0 | 36.7±1.4 | 56.7±7.3 | 61.7±1.7 | 66.7±1.7 | **80.0±2.9** |
| | | 10% | 8.3±1.7 | 36.7±1.7 | 48.3±1.7 | 61.7±1.7 | 61.7±1.7 | 0.0±0.0 | 58.3±4.4 | 60.0±0.0 | 65.0±2.9 | **73.3±1.7** |
| | | 30% | 10.0±0.0 | 28.3±1.7 | 46.7±1.7 | 63.3±1.7 | 60.0±0.0 | 40.0±2.4 | 35.0±2.9 | 53.3±1.7 | 50.0±0.0 | **76.7±1.7** |
| | | 50% | 0.0±0.0 | 38.3±1.7 | 45.0±0.0 | 65.0±0.0 | 53.3±1.7 | 0.0±0.0 | 53.3±1.7 | 46.7±1.7 | 45.0±0.0 | **78.3±4.4** |
| | | 70% | 0.0±0.0 | 55.0±0.0 | 38.3±3.3 | 65.0±0.0 | 50.0±0.0 | 26.7±5.4 | 30.0±5.8 | 33.3±1.7 | 33.3±1.7 | **66.7±1.7** |
| | | 90% | 0.0±0.0 | 21.7±1.7 | 26.7±1.7 | 46.7±1.7 | 3.3±1.7 | 0.0±0.0 | 6.7±3.3 | 8.3±1.7 | 15.0±0.0 | **53.3±1.7** |
| | LP-variant | 0% | 48.3±3.3 | 40.0±2.9 | 41.7±1.7 | 65.0±0.0 | 58.3±1.7 | 30.0±0.0 | 61.7±1.7 | 75.0±2.9 | 71.7±1.7 | **75.0±0.0** |
| | | 10% | 0.0±0.0 | 36.7±1.7 | 40.0±0.0 | 63.3±3.3 | 60.0±0.0 | 36.7±2.7 | 65.0±2.9 | 70.0±2.9 | 73.3±1.7 | **80.0±2.9** |
| | | 30% | 0.0±0.0 | 48.3±3.3 | 40.0±2.9 | 60.0±0.0 | 55.0±0.0 | 40.0±7.1 | 53.3±6.0 | 65.0±2.9 | 65.0±0.0 | **75.0±0.0** |
| | | 50% | 0.0±0.0 | 65.0±0.0 | 35.0±2.9 | 63.3±3.3 | 60.0±0.0 | 38.3±3.6 | 48.3±4.4 | 61.7±1.7 | 65.0±0.0 | **78.3±1.7** |
| | | 70% | 0.0±0.0 | 46.7±3.3 | 35.0±0.0 | 70.0±0.0 | 60.0±0.0 | 25.0±8.2 | 60.0±5.0 | 56.7±1.7 | 56.7±1.7 | **71.7±1.7** |
| | | 90% | 0.0±0.0 | 35.0±2.9 | 50.0±0.0 | 65.0±2.9 | 0.0±0.0 | 30.0±12.5 | 50.0±2.9 | 38.3±1.7 | 55.0±2.9 | **66.7±1.7** |
| Re-mapped label tasks | AG News Remap | 0% | 20.0±2.9 | 15.0±0.0 | 26.7±1.7 | 43.3±1.7 | 43.3±3.3 | 5.0±2.4 | 25.0±5.0 | 40.0±0.0 | 40.0±0.0 | **50.0±0.0** |
| | | 10% | 5.0±0.0 | 15.0±0.0 | 15.0±0.0 | 41.7±1.7 | 38.3±1.7 | 3.3±1.4 | 26.7±3.3 | 36.7±1.7 | 40.0±0.0 | **51.7±1.7** |
| | | 30% | 10.0±0.0 | 5.0±0.0 | 5.0±0.0 | 40.0±0.0 | 36.7±1.7 | 1.7±1.4 | 10.0±0.0 | 40.0±0.0 | 43.3±1.7 | **55.0±0.0** |
| | | 50% | 5.0±0.0 | 10.0±0.0 | 5.0±0.0 | 43.3±1.7 | 35.0±0.0 | 3.3±1.4 | 20.0±5.0 | 35.0±0.0 | 35.0±0.0 | **55.0±2.9** |
| | | 70% | 5.0±0.0 | 25.0±0.0 | 8.3±1.7 | 50.0±0.0 | 35.0±0.0 | 1.7±1.4 | 11.7±6.7 | 38.3±1.7 | 46.7±1.7 | **58.3±3.3** |
| | | 90% | 5.0±0.0 | 18.3±1.7 | 5.0±0.0 | 40.0±0.0 | 10.0±0.0 | 15.0±6.2 | 35.0±0.0 | 35.0±0.0 | 41.7±1.7 | **53.3±1.7** |
| | SST5 Reverse | 0% | 20.0±0.0 | 10.0±0.0 | 13.3±1.7 | 40.0±0.0 | 40.0±0.0 | 15.0±2.4 | 33.3±6.7 | 35.0±2.9 | 40.0±0.0 | **50.0±2.9** |
| | | 10% | 16.7±1.7 | 10.0±0.0 | 15.0±0.0 | **48.3±1.7** | 40.0±0.0 | 13.3±2.7 | 23.3±6.7 | 33.3±3.3 | 40.0±0.0 | **48.3±1.7** |
| | | 30% | 23.3±1.7 | 6.7±1.7 | 25.0±2.9 | 40.0±0.0 | 40.0±0.0 | 21.7±3.6 | 26.7±1.7 | 30.0±0.0 | 31.7±1.7 | **46.7±3.3** |
| | | 50% | 21.7±1.7 | 15.0±0.0 | 15.0±0.0 | 43.3±1.7 | 33.3±1.7 | 21.7±1.4 | 23.3±1.7 | 28.3±1.7 | 30.0±0.0 | **46.7±3.3** |
| | | 70% | 25.0±0.0 | 23.3±1.7 | 23.3±1.7 | 40.0±0.0 | 30.0±0.0 | 20.0±2.4 | 25.0±2.9 | 36.7±1.7 | 36.7±1.7 | **45.0±5.0** |
| | | 90% | 20.0±0.0 | 15.0±2.9 | 20.0±0.0 | 30.0±0.0 | 30.0±0.0 | 13.3±2.7 | 21.7±1.7 | 30.0±0.0 | 30.0±0.0 | **31.7±1.7** |

Table 3: Average accuracy ± s.e. for EASE with and without jointly optimized instructions. We removed tasks with 100% accuracy. The full results are in App. C, Tab. 6.

| | EASE | EASE with instructions | Improvement |
|---|---|---|---|
| antonyms | **90.0±0.0** | 85.0±0.0 | -5.0 ↓ |
| auto_categorization | 30.0±0.0 | **56.7±1.7** | 26.7 ↑ |
| negation | 95.0±0.0 | **100.0±0.0** | 5.0 ↑ |
| object_counting | 73.3±1.7 | **75.0±0.0** | 1.7 ↑ |
| orthography_starts_with | 80.0±0.0 | **81.7±1.7** | 1.7 ↑ |
| second_word_letter | 53.3±1.7 | **100.0±0.0** | 46.7 ↑ |
| sentence_similarity | 56.7±1.7 | **58.3±1.7** | 1.7 ↑ |
| synonyms | 30.0±0.0 | **31.7±1.7** | 1.7 ↑ |
| taxonomy_animal | 88.3±1.7 | **100.0±0.0** | 11.7 ↑ |
| translation_en-de | **90.0±0.0** | **90.0±0.0** | 0.0 ○ |
| translation_en-fr | **88.3±1.7** | 85.0±0.0 | -3.3 ↓ |
| word_sorting | 90.0±0.0 | **93.3±1.7** | 3.3 ↑ |
| word_unscrambling | 78.3±1.7 | **80.0±0.0** | 1.7 ↑ |
| LR (10% noise) | **73.3±1.7** | 45.0±15.0 | -28.3 ↓ |
| LP-variant (10% noise) | 80.0±2.9 | **86.7±1.7** | 6.7 ↑ |
| AG News Remap (10% noise) | 51.7±1.7 | **65.0±0.0** | 13.3 ↑ |
| SST5 Reverse (10% noise) | 48.3±1.7 | **53.3±1.7** | 5.0 ↑ |

Table 4: Average accuracy ± s.e. achieved by EASE and EASE with retrieval for larger exemplar set sizes.

**AG News Remap (10% noise)**

| Size $n$ | EASE | EASE with retrieval |
|---|---|---|
| 1000 | 50.0±2.9 | **60.0±0.0** |
| 10000 | 55.0±0.0 | **63.3±1.7** |
| 50000 | 48.3±1.7 | **63.3±1.7** |
| 100000 | 50.0±2.9 | **65.0±0.0** |

**SST5 Reverse (10% noise)**

| Size $n$ | EASE | EASE with retrieval |
|---|---|---|
| 1000 | 43.3±3.3 | **48.3±1.7** |
| 3000 | 48.3±4.4 | **48.3±1.7** |
| 5000 | 43.3±1.7 | **50.0±0.0** |
| 7000 | 45.0±0.0 | **48.3±1.7** |

those that are less relevant to the task. This independent step to pre-filter candidates serves as a promising extension of our EASE to handle a large set of exemplars. To this end, we propose to first use retrieval-based methods to select the exemplars that are more relevant to the task, and then run our EASE using this refined smaller set of exemplars. Specifically, we use a cosine similarity retriever on exemplar embedding and perform exemplar selection on $D$ with a size $n$ as large as 100000. The query involves validation data exemplars $D_V$ and we retrieve from the corpus of training set exemplars $D$. For each validation data exemplar in $D_V$, we retrieve the most similar/relevant training set exemplars from $D$. Then, we combine them to form a smaller set of exemplars than $D$ and proceed with EASE using this reduced subset for efficiency. As shown in Tab. 4, when the size $n$ of the exemplar set is large, combining EASE with retrieval gives better performances than directly running EASE.

**Effectiveness of components of EASE.** Here we show the necessity of both of the main components to the success of EASE: OT and NeuralUCB. As shown in Tab. 7 of App. C.3, EASE significantly outperforms methods employing OT or NeuralUCB alone. Overall, a pure subset selection algorithm based on OT performs badly on its own, especially when the dataset's noise ratio is high. However, when used together with NeuralUCB, OT significantly improves the performance of our EASE.


https://doi.org/10.52202/079017-3899

**Further ablations.** We defer further ablation studies to App. C, including those exploring (a) speedups from OT, (b) a larger $k$ or a range of $k$, (c) a larger query budget $T$, (d) benefits of using an ordering-aware embedding, (e) different black-box LLMs, (f) other embedding models, (g) degree of exploration $\nu$, (h) asking GPT to directly select exemplars, and (i) additional experiments on more datasets including reasoning chains.

## 5 Related work

**Retrieval-based methods.** This approach utilizes trainable exemplar retrievers to select exemplars depending on the text sequence of each individual test sample [1, 48]. Liu et al. [23] first proposed a similarity-based (e.g., negative Euclidean distance or cosine similarity) retrieval strategy on the sentence embeddings of the exemplars for ICL. Gao et al. [11] further enhanced the above by emphasizing on exemplars with top-2 labels that the LLM is most uncertain about. Adapting retrievers to specific tasks, Rubin et al. [38] learned a retrieval model from evaluating individual exemplars by their 1-shot ICL performance on validation data. Improving on the limitation of the above methods that exemplars are considered in isolation (i.e., independently), Ye et al. [48] proposed to retrieve a set of exemplars jointly by examining their joint probability to capture inter-relationships. Likewise, Levy et al. [18] retrieved a set by selecting diverse exemplars that collectively cover all of the output structures. Gupta et al. [14] instead generalized the individual metrics to a set-level metric through a submodular function, which is well-suited for optimization via greedy algorithms. However, retrieval-based methods are characterized by varying exemplars for each test sample, which do not align with our practical setting of maintaining a fixed set of exemplars for the entire task. Also, another common drawback is that heuristics are typically required to order the exemplars in the retrieved set.

**Selection of a fixed set of exemplars.** Having a fixed set of exemplars offers practical and privacy-related advantages, such as the ease of implementation and reduced data exposure [25, 30]. Wang et al. [44] proposed to train a small LLM as a latent variable model using output token logits from independent exemplars, then forming a fixed exemplar set to directly transfer to target models. Similarly, Chang and Jia [4] developed a data model for each validation sample and introduced an aggregate metric to select subsets. Li and Qiu [19] proposed to filter and search for best exemplars using an informativeness metric derived from LLM output logits on classification tasks. However, these works are restricted to classification tasks. The closest to our work is that of Zhang et al. [50], which used reinforcement learning to actively select exemplars. In a similar setting to ours, Nguyen and Wong [31] used influence to select the most influential exemplars and order them arbitrarily to form the subset. However, both methods perform worse than our algorithm in our experiments.

**Instruction optimization.** Another related direction for enhancing the performance of LLMs is instruction optimization. Focusing on instructions within the prompt, evolutionary algorithms and zeroth-order optimization algorithms are gaining popularity in refining the prompts for black-box LLMs [5, 10, 13, 16, 21, 47]. Specifically, some studies [10, 16, 21, 47] maintained a constant set of exemplars throughout the process of prompt optimization, whereas others [5, 9, 13, 34] ignored the consideration of exemplars altogether by adopting a zero-shot setting. It is therefore imperative to develop an effective exemplar selection method for black-box LLMs.

## 6 Conclusion and limitation

We propose `EASE`, an algorithm that selects the optimal ordered set of exemplars for in-context learning of black-box LLMs in an automated fashion. `EASE` is query-efficient due to the adoption of the NeuralUCB algorithm and is further made computationally feasible for large spaces of exemplar sequences through a technique based on optimal transport. Additionally, our `EASE` has been extended to a fully automated prompt optimization pipeline that jointly optimizes exemplars and instruction for the best in-context learning performance. Furthermore, we provide practical insights indicating that exemplar selection in in-context learning is more crucial for downstream tasks that the LLM has limited knowledge about. However, the on-the-fly computation of embedding for the ordered exemplar sequences is the computational bottleneck of our method, which could be potentially improved in future work for more efficient optimization. Also, a potential limitation of `EASE` is the requirement for a suitable validation set, which may not be readily available in some scenarios.

## Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation Singapore and the Singapore Ministry of Digital Development and Innovation, National AI Group under the AI Visiting Professorship Programme (award number AIVP-2024-001). This research is supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The Mens, Manus, and Machina (M3S) is an interdisciplinary research group (IRG) of the Singapore MIT Alliance for Research and Technology (SMART) centre.

## References

[1] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *arXiv:2402.16827*, 2024.

[2] Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *Management Science*, 67(3):1329–1349, 2021.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proc. NeurIPS*, pages 1877–1901, 2020.

[4] Ting-Yun Chang and Robin Jia. Data curation alone can stabilize in-context learning. In *Proc. ACL*, pages 8123–8144, 2023.

[5] Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. InstructZero: Efficient instruction optimization for black-box large language models. *arXiv:2306.03082*, 2023.

[6] Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time? *arXiv:2307.09009*, 2023.

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

[8] Wendi Cui, Jiaxin Zhang, Zhuohang Li, Hao Sun, Damien Lopez, Kamalika Das, Bradley Malin, and Sricharan Kumar. Phaseevo: Towards unified in-context prompt optimization for large language models. *arXiv:2402.11347*, 2024.

[9] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proc. EMNLP*, pages 3369–3391, 2022.

[10] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv:2309.16797*, 2023.

[11] Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. Ambiguity-aware in-context learning with large language models. *arXiv:2309.07900*, 2024.

[12] Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik,

 https://doi.org/10.52202/079017-3899

Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In *Proc. ACL*, pages 15789–15809, 2024.

[13] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *Proc. ICLR*, 2024.

[14] Shivanshu Gupta, Matt Gardner, and Sameer Singh. Coverage-based example selection for in-context learning. In *Proc. EMNLP*, pages 13924–13950, 2023.

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proc. NeurIPS (Track on Datasets and Benchmarks)*, 2021.

[16] Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiangqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. Localized zeroth-order prompt optimization. *arXiv:2403.02993*, 2024.

[17] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *Proc. ICLR*, 2024.

[18] Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In *Proc. ACL*, pages 1401–1422, 2023.

[19] Xiaonan Li and Xipeng Qiu. Finding support examples for in-context learning. In *Proc. EMNLP*, pages 6219–6235, 2023.

[20] Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *Proc. ACL*, pages 4644–4668, 2023.

[21] Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Use your INSTINCT: Instruction optimization using neural bandits coupled with transformers. In *NeurIPS Workshop on Instruction Tuning and Instruction Following*, 2023.

[22] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proc. ACL*, pages 158–167, 2017.

[23] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proc. DeeLIO: Deep Learning Inside Out*, pages 100–114, 2022.

[24] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. ACL*, pages 8086–8098, 2022.

[25] Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey. *arXiv:2401.11624*, 2024.

[26] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. The AI index 2024 annual report. Technical report, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University, 2024.

[27] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proc. EMNLP*, pages 11048–11064, 2022.

[28] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv:2402.06196*, 2024.

[29] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Proc. ACL Findings*, pages 12284–12314, 2023.

[30] Seth Neel and Peter Chang. Privacy issues in large language models: A survey. *arXiv:2312.06717*, 2024.

[31] Tai Nguyen and Eric Wong. In-context example selection with influences. *arXiv:2302.11042*, 2023.

[32] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, and Sam Altman et al. GPT-4 technical report. *arXiv:2303.08774*, 2024.

[33] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. In *Proc. NeurIPS*, pages 11054–11070, 2021.

[34] Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proc. EACL*, pages 3845–3864, 2023.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, pages 8748–8763, 2021.

[36] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proc. EMNLP*, pages 3982–3992, 2019.

[37] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[38] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proc. NAACL*, pages 2655–2671, 2022.

[39] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41(5): 2263–2291, 2013.

[40] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: masked and permuted pre-training for language understanding. In *Proc. NeurIPS*, pages 16857–16867, 2020.

[41] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, and Katie Millican et al. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2024.

[42] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

[43] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proc. NeurIPS*, pages 5776–5788, 2020.

[44] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Proc. NeurIPS*, pages 15614–15638, 2023.

[45] Jeremy White. How strangers got my email address from ChatGPT's model. *The New York Times*, 2023. URL `https://www.nytimes.com/interactive/2023/12/22/technology/openai-chatgpt-privacy-exploit.html`.

 https://doi.org/10.52202/079017-3899

[46] Xinyi Xu, Zhaoxuan Wu, Rui Qiao, Arun Verma, Yao Shu, Jingtan Wang, Xinyuan Niu, Zhenfeng He, Jiangwei Chen, Zijian Zhou, Gregory Kang Ruey Lau, Hieu Dao, Lucas Agussurja, Rachael Hwee Ling Sim, Xiaoqiang Lin, Wenyang Hu, Zhongxiang Dai, Pang Wei Koh, and Bryan Kian Hsiang Low. Data-centric ai in the age of large language models. In *Proc. EMNLP Findings*, 2024.

[47] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *Proc. ICLR*, 2024.

[48] Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. In *Proc. ICML*, pages 39818–39833, 2023.

[49] Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. Tempera: Test-time prompting via reinforcement learning. In *Proc. ICLR*, 2023.

[50] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proc. EMNLP*, pages 9134–9148, 2022.

[51] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *arXiv:2303.18223*, 2023.

[52] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proc. ICML*, pages 12697–12706, 2021.

[53] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In *Proc. ICML*, pages 11492–11502, 2020.

[54] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *Proc. ICLR*, 2023.

# A Broader impacts

As LLMs continue to gain popularity and are increasingly used by a broad user base (e.g., evidenced by OpenAI's over 1 million active users), carefully handling the ethical considerations associated with LLMs' diverse applications becomes crucial. Such a work like ours which focuses on automatically optimizing the performance of LLMs offers valuable applications and ease of usage. However, it is both challenging and essential to address the potential for malicious usage. When applied to tasks specially designed with malicious or adversarial intent, our method could be exploited to produce harmful instructions and exemplars. In such cases, responsible usage of the tool is important. There may be a need for a user-friendly platform (which integrates safety measures, instead of providing the raw code) to add an extra layer of protection. Additionally, we urge the community to focus more on potential ethical and safety issues related to the usage of LLMs and associated technologies, which should also account for additional objectives and constraints (e.g., harmfulness).

# B Implementation details

In this section, we provide details for the implementation of the data processing (see App. B.1) and the baseline methods employed in this paper (see App. B.2). All experiments are conducted on a server with Intel(R) Xeon(R) CPU and NVIDIA H100 GPUs. Unless otherwise stated, we use "gpt-3.5-turbo-1106" as the target black-box model, and MPNet as the embedding model.

## B.1 Details for data processing

We follow the query template proposed by APE [54] for querying LLM models. Example 1 below shows the general query template for in-context learning (ICL) used in the paper, and the LLM will generate outputs corresponding to the query. In the template, the placeholders (i.e., [INSTRUCTION], [INPUT], [OUTPUT] and [TEST INPUT]) are replaced by raw text in the datasets. The placeholder "<More exemplars...>" represents any additional exemplars written the same input-output format depending on the number of in-context exemplars $k$ in the prompt. The "instruction" is optional in the query and we only include the instruction when jointly optimizing for both the exemplars and the instruction (Sec. 3.3 and Sec. 4.4).

**Example Query 1: A General Template**

*(optional)* Instruction: [INSTRUCTION]

Input: [INPUT]
Output: [OUTPUT]

<More exemplars...>

Input: [INPUT]
Output: [OUTPUT]

Input: [TEST INPUT]
Output:

Next, we present the details for the new families of out-of-distribution tasks that we proposed in Sec. 4.3.

**Linear regression (LR)**. The LR task is generated with an underlying linear regression function $y = ax + b$, where $a, b$ are the coefficients to be defined. In our specific example of the LR task presented in the paper, we arbitrarily choose $a = -4$ and $b = 6$. We let $x$ be the input and $y$ be the output. Note that to make the task difficult, no information about the function structure (i.e., $y = ax + b$) is passed to the LLM in the query. A concrete example is shown in Example 2.

**Language puzzle variant (LP-variant)**. We change the rules for the classic language game "pig Latin". The original pig Latin follows 2 rules: (a) For words that begin with a vowel, one just adds "yay" to the end; (b) For words that begin with consonants, the initial consonants are moved to the end

 https://doi.org/10.52202/079017-3899

of the word, then "ay" is added. We create a variant, LP-variant, using the "ay" suffix for both rules (a) and (b). An example of the task query is given in Example 3. Note that one could freely modify the rules, by changing the suffix word "yay" to something else for example, and create diverse tasks that require in-context reasoning from the model.

**Example Query 2: LR**

Input: 172
Output: -682

<More exemplars...>

Input: 47
Output: -182

Input: 117
Output:

**Example Query 3: LP-variant**

Input: quick brown fox The jumps Over the Lazy dog
Output: uickqay ownbray oxfay Ethay umpsjay Overyay ethay Azylay ogday

<More exemplars...>

Input: Tom never walks to school
Output: Omtay evernay alksway otay oolschay

Input: We never talk
Output:

**AG News Remap**. For the commonly-used AG News dataset, the news articles are classified into four categories: "World", "Sports", "Business" and "Sci/Tech". We constructed a re-mapped dataset by "shifting" the label mapping, such that

- "World" news is now labeled as "Sports" news,
- "Sports" news is now labeled as "Business" news,
- "Business" news is now labeled as "Sci/Tech" news,
- "Sci/Tech" news is now labeled as "World" news.

Note that the remapping rule above is chosen arbitrarily. Therefore, one could create a variety of tasks by changing the remapping rule, or even directly changing the labels completely. For the above rule that we adopted, an example of the AG News Remap task query is given in Example 4.

**Example Query 4: AG News Remap**

Input: Yahoo, Adobe join hands Adobe Systems, the digital imaging, design and document technology platform provider and Internet service provider Yahoo will announce this week the launch of a co-branded Yahoo Toolbar.
Output: World

<More exemplars...>

Input: Schumacher Sets Mark Michael Schumacher won the Hungarian Grand Prix Sunday in Budapest, setting yet another record by becoming the first Formula One driver with 12 victories in a season.
Output: Business

Input: LeapFrog Warns on 3Q, Year Profit View LeapFrog Enterprises Inc., a developer of technology-based educational products, on Monday lowered third-quarter and full-year profit expectations, citing difficult market conditions.
Output:

**SST5 Reverse**. SST5 is another commonly used 5-way sentiment classification dataset, with labels being "very positive", "positive", "neutral", "negative" and "very negative". To make the task novel and unseen by the LLM before, we reverse the labels to be against the sentiment expressed in the input, such that

- "very positive" sentences are now labeled as "very negative",
- "positive" sentences are now labeled as "negative",

Table 5: Average accuracy ± standard error achieved by the best exemplar sequence discovered by different algorithms for different tasks over 3 independent trials.

| | DPP | MMD | OT | Cosine | BM25 | Active | Inf | Evo | Best-of-N | EASE |
|---|---|---|---|---|---|---|---|---|---|---|
| active_to_passive | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| antonyms | 70.0$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | 81.7$_{\pm 1.7}$ | 85.0$_{\pm 0.0}$ | 85.0$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | 86.7$_{\pm 1.7}$ | 88.3$_{\pm 1.7}$ | **90.0**$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ |
| auto_categorization | 3.3$_{\pm 1.7}$ | 8.3$_{\pm 1.7}$ | 0.0$_{\pm 0.0}$ | 25.0$_{\pm 0.0}$ | 16.7$_{\pm 1.7}$ | 10.0$_{\pm 2.4}$ | 21.7$_{\pm 1.7}$ | 21.7$_{\pm 1.7}$ | 20.0$_{\pm 0.0}$ | **30.0**$_{\pm 0.0}$ |
| diff | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| first_word_letter | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| larger_animal | 70.0$_{\pm 0.0}$ | 91.7$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 66.7$_{\pm 1.4}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| letters_list | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| negation | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ |
| num_to_verbal | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| object_counting | 55.0$_{\pm 2.9}$ | 56.7$_{\pm 1.7}$ | 48.3$_{\pm 1.7}$ | 61.7$_{\pm 1.7}$ | 66.7$_{\pm 1.7}$ | 51.7$_{\pm 1.4}$ | 63.3$_{\pm 4.4}$ | 70.0$_{\pm 0.0}$ | 70.0$_{\pm 0.0}$ | **73.3**$_{\pm 1.7}$ |
| orthography_starts_with | 20.0$_{\pm 2.9}$ | 35.0$_{\pm 0.0}$ | 61.7$_{\pm 1.7}$ | 78.3$_{\pm 1.7}$ | 70.0$_{\pm 0.0}$ | 43.3$_{\pm 1.4}$ | 70.0$_{\pm 2.9}$ | 75.0$_{\pm 0.0}$ | 78.3$_{\pm 1.7}$ | **80.0**$_{\pm 0.0}$ |
| rhymes | 60.0$_{\pm 0.0}$ | 51.7$_{\pm 1.7}$ | 0.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | 65.0$_{\pm 8.2}$ | 70.0$_{\pm 13.2}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| second_word_letter | 10.0$_{\pm 2.9}$ | 30.0$_{\pm 0.0}$ | 28.3$_{\pm 1.7}$ | 50.0$_{\pm 0.0}$ | 50.0$_{\pm 0.0}$ | 26.7$_{\pm 8.3}$ | 40.0$_{\pm 0.0}$ | 46.7$_{\pm 1.7}$ | 50.0$_{\pm 0.0}$ | **53.3**$_{\pm 1.7}$ |
| sentence_similarity | 20.0$_{\pm 0.0}$ | 21.7$_{\pm 3.3}$ | 40.0$_{\pm 2.9}$ | 46.7$_{\pm 1.7}$ | 53.3$_{\pm 1.7}$ | 5.0$_{\pm 4.1}$ | 18.3$_{\pm 6.7}$ | 45.0$_{\pm 0.0}$ | 51.7$_{\pm 1.7}$ | **56.7**$_{\pm 1.7}$ |
| sentiment | 85.0$_{\pm 0.0}$ | 90.0$_{\pm 0.0}$ | 85.0$_{\pm 0.0}$ | 96.7$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | 85.0$_{\pm 4.1}$ | 91.7$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| singular_to_plural | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| sum | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| synonyms | 10.0$_{\pm 0.0}$ | 25.0$_{\pm 0.0}$ | 20.0$_{\pm 0.0}$ | **35.0**$_{\pm 0.0}$ | 30.0$_{\pm 0.0}$ | 3.3$_{\pm 1.4}$ | 26.7$_{\pm 1.7}$ | 30.0$_{\pm 0.0}$ | 30.0$_{\pm 0.0}$ | 30.0$_{\pm 0.0}$ |
| taxonomy_animal | 43.3$_{\pm 4.4}$ | 40.0$_{\pm 2.9}$ | 46.7$_{\pm 1.7}$ | 85.0$_{\pm 2.9}$ | 80.0$_{\pm 0.0}$ | 45.0$_{\pm 6.2}$ | 70.0$_{\pm 5.0}$ | 80.0$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | **88.3**$_{\pm 1.7}$ |
| translation_en-de | **90.0**$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | 80.0$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ | 85.0$_{\pm 0.0}$ | 56.7$_{\pm 13.0}$ | **90.0**$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ |
| translation_en-es | 90.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 96.7$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 96.7$_{\pm 1.4}$ | 98.3$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| translation_en-fr | 76.7$_{\pm 1.7}$ | 76.7$_{\pm 1.7}$ | 81.7$_{\pm 1.7}$ | 85.0$_{\pm 0.0}$ | 85.0$_{\pm 0.0}$ | 81.7$_{\pm 1.4}$ | 85.0$_{\pm 0.0}$ | 86.7$_{\pm 1.7}$ | 85.0$_{\pm 0.0}$ | **88.3**$_{\pm 1.7}$ |
| word_sorting | 26.7$_{\pm 1.7}$ | 88.3$_{\pm 1.7}$ | 88.3$_{\pm 1.7}$ | 90.0$_{\pm 0.0}$ | 71.7$_{\pm 1.7}$ | 80.0$_{\pm 0.0}$ | 88.3$_{\pm 1.7}$ | **93.3**$_{\pm 1.7}$ | 91.7$_{\pm 1.7}$ | 90.0$_{\pm 0.0}$ |
| word_unscrambling | 68.3$_{\pm 1.7}$ | 56.7$_{\pm 1.7}$ | 71.7$_{\pm 1.7}$ | 75.0$_{\pm 0.0}$ | 76.7$_{\pm 1.7}$ | 63.3$_{\pm 3.6}$ | 66.7$_{\pm 1.7}$ | 75.0$_{\pm 0.0}$ | 75.0$_{\pm 0.0}$ | **78.3**$_{\pm 1.7}$ |
| # best-performing tasks | 7 | 7 | 7 | 11 | 9 | 6 | 10 | 14 | 14 | **22** |

- "neutral" sentences are still labeled as "neutral",
- "negative" sentences are now labeled as "positive",
- "very negative" sentences are now labeled as "very positive".

This makes the task difficult as the LLM has to now output sentiments that are against the pre-trained knowledge about sentiments, which is obtained from the large corpus of pre-training data that the LLM has been trained on.

An example of the SST5 Reverse task query is given in Example 5.

**Example Query 5: SST5 Reverse**

Input: extremely bad .
Output: very positive

<More exemplars...>

Input: ... bright , intelligent , and humanly funny film .
Output: very negative

Input: really dumb but occasionally really funny .
Output:

**Injecting noises into datasets.** To construct noisy datasets with different noise ratios, we mainly modify the labels of data points. Specifically, to construct a dataset with $r\%$ noisy data, we sample $r\%$ data and replace their labels with the labels of other randomly sampled data points in the dataset. We also design specific noise structures for LR and LP-variant to simulate noises due to a systematic error. For LR, the noisy data instead follows $y = 5x - 8$. For LP-variant, the noisy data simply repeats the input sentence in the label.

### B.2 Details for implementation of the methods and baselines

For all methods, we use a consistent exemplar selection set $D$, validation set $D_V$ and test set for evaluating one task. We also implement all exemplar selection without replacement for all methods.


https://doi.org/10.52202/079017-3899

**DDP**. We follow Ye et al. [48] to use the DPP metric combined with relevance scores as the subset selection criterion. We do not train the embedding model and directly use cosine similarity to measure the relevance term. However, the original metric by Ye et al. [48] is dependent on the test input $x_{\text{test}}$, so we adapted the method to average the metric over the validation dataset. The exemplar set with the maximum values on the metric is chosen and evaluated on the black-box LLM.

**MMD**. We use the MMD metric in Sejdinovic et al. [39] to measure the distance between the exemplars set and the validation set. The exemplar set with the maximum values on the MMD metric is chosen and evaluated on the black-box LLM.

**OT**. We use the OT metric in (4) to measure the distance between the exemplars set and the validation set. Similarly, the exemplar set with the maximum values on the OT metric is chosen and evaluated on the black-box LLM.

**Cosine**. Retrieval-based methods are not suitable for finding a fixed set of exemplars for the entire test set in its original form. Hence, we propose the following adaptation. We first retrieve top-$R$ candidates with the lowest distance to all validation samples on average. Then, we sample $T$ permutations of exemplars from the $R$ candidates to test on the black-box LLM, where $T$ is the black-box query budget. In the above, the retriever calculates the cosine similarity between the embeddings of samples obtained from an embedding model $h(\cdot)$. In this paper, we use $R = 10$ and use MPNet as $h(\cdot)$.

**BM25**. This retrieval-based baseline works similarly to Cosine. The only difference is that this baseline uses the classic BM25 [37] as the retriever.

**Active**. We use the official implementation of Zhang et al. [50]. For a fair comparison using the same query budget $T$ for the black-box LLM, we limit the number of episodes that can be used according to $T$, and then train the exemplar selection policy through reinforcement learning.

**Inf**. For Inf, we first sample random permutations of exemplar sequences to be evaluated on the black-box LLM to obtain the scores. Then, we follow the influence metric proposed by Nguyen and Wong [31] to select $k$ individual exemplars to form the best exemplar sequence. The best exemplar sequence is then finally evaluated on the black-box LLM.

**Evo.** We propose a new baseline using evolutionary strategies. For each iteration, the mutation operator changes one exemplar in the current best exemplar sequence to another random exemplar. Then, we evaluate the exemplars on the black-box LLM. In this way, we exploit the current knowledge about the best exemplars and perform local mutations.

**Best-of-N.** This is a straightforward baseline that explores the whole space of all exemplar permutations uniformly by sampling random permutations until the $T$ query budget is exhausted.

`EASE`. The details for our proposed method `EASE` are described thoroughly in Sec. 3. We use a sampling size of $q = 50000$ exemplar permutations per iteration after OT is introduced.

# C More experiment results

## C.1 Full results table for the 24 tasks

Tab. 5 is the full table containing all 24 tasks (with more than 100 training data samples) from the Instruction Induction dataset (compared to Tab. 1 which drops tasks with 100% accuracy across baselines). Our `EASE` outperforms all baselines.

## C.2 Full results table for `EASE` with instructions

We present the full table of results for `EASE` with joint optimization of exemplars and instructions in Tab. 6 (compared to Tab. 3 which drops tasks with 100% accuracy both with and without instructions). The conclusion is still consistent with the main text that incorporating instructions jointly optimized to best match the chosen exemplars could improve the ICL performance, especially for difficult tasks (e.g., auto categorization, taxonomy animal, etc.).

We also provide some details on the efficient implementation of the joint optimization in practice. According to Line 5 of Algorithm 1, the instructions augment the search space by the size $|P|$. This will gain us much benefit if additional computation is available. Otherwise, in practice, the most straightforward implementation without increasing the computational complexity is to reduce

Table 6: Average accuracy ± standard error comparison for EASE with and without jointly optimized instructions.

| | EASE | EASE with instructions | improvement |
|---|---|---|---|
| antonyms | **90.0**$_{\pm 0.0}$ | 85.0$_{\pm 0.0}$ | -5.0 ↓ |
| auto_categorization | 30.0$_{\pm 0.0}$ | **56.7**$_{\pm 1.7}$ | 26.7 ↑ |
| diff | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| larger_animal | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| negation | 95.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 5.0 ↑ |
| object_counting | 73.3$_{\pm 1.7}$ | **75.0**$_{\pm 0.0}$ | 1.7 ↑ |
| orthography_starts_with | 80.0$_{\pm 0.0}$ | **81.7**$_{\pm 1.7}$ | 1.7 ↑ |
| rhymes | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| second_word_letter | 53.3$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | 46.7 ↑ |
| sentence_similarity | 56.7$_{\pm 1.7}$ | **58.3**$_{\pm 1.7}$ | 1.7 ↑ |
| sentiment | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| sum | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| synonyms | 30.0$_{\pm 0.0}$ | **31.7**$_{\pm 1.7}$ | 1.7 ↑ |
| taxonomy_animal | 88.3$_{\pm 1.7}$ | **100.0**$_{\pm 0.0}$ | 11.7 ↑ |
| translation_en-de | **90.0**$_{\pm 0.0}$ | **90.0**$_{\pm 0.0}$ | 0.0 ○ |
| translation_en-es | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| translation_en-fr | **88.3**$_{\pm 1.7}$ | 85.0$_{\pm 0.0}$ | -3.3 ↓ |
| word_sorting | 90.0$_{\pm 0.0}$ | **93.3**$_{\pm 1.7}$ | 3.3 ↑ |
| word_unscrambling | 78.3$_{\pm 1.7}$ | **80.0**$_{\pm 0.0}$ | 1.7 ↑ |
| LR (10% noise) | **73.3**$_{\pm 1.7}$ | 45.0$_{\pm 15.0}$ | -28.3 ↓ |
| LP-variant (10% noise) | 80.0$_{\pm 2.9}$ | **86.7**$_{\pm 1.7}$ | 6.7 ↑ |
| AG News Remap (10% noise) | 51.7$_{\pm 1.7}$ | **65.0**$_{\pm 0.0}$ | 13.3 ↑ |
| SST5 Reverse (10% noise) | 48.3$_{\pm 1.7}$ | **53.3**$_{\pm 1.7}$ | 5.0 ↑ |

$|Q'_t|$ from $q'$ to $q'/|P|$. More generally, one can randomly sample $r \times |P|$ instructions to pair with $q'/(r \times |P|)$ exemplar sequences. This implementation is simple yet effective, where $r$ controls the trade-off of focusing more on instructions versus exemplars.

### C.3 Ablation studies for the NeuralUCB and OT components

The following Tab. 7 and Tab. 8 show the necessity of both of the main components, NeuralUCB and OT, in the success of the proposed EASE algorithm. A pure subset selection algorithm based on OT performs badly on its own, especially when the noise ratio is high in the datasets. However, it greatly improves the performance of NeuralUCB when used together with NeuralUCB in our EASE. This is attributed to the ability to examine more permutations of the exemplars without increasing the computational cost, since only selected permutations through OT (which places an implicit bias towards more relevant exemplars) will be evaluated for the acquisition values in NeuralUCB.

### C.4 Ablation studies for the speedups from OT

Carrying on from the previous section about the necessity of both the NeuralUCB and OT components, we can quantify through experiments the amount of speedups brought by OT. Specifically, we measure the wall clock time speedups for different sizes of the domain space $Q_t$ that we sample. We experiment on one of the tasks with the longest input, AG News Remap. As shown in Tab. 9, the larger the domain space $|Q_t|$, the greater the relative gain in the speedups. This mitigates the computational issues of applying NeuralUCB to the problem of exemplar selection and contributes positively to the success of EASE.

### C.5 Ablation studies for other numbers of $k$

We validate that EASE is able to select a larger set of exemplars (i.e., larger $k$) and also from a larger exemplar set of size $n$. We conduct comparisons of EASE to the two most competitive baselines, Evo and Best-of-N, on the AG News Remap and SST5 Reverse datasets of size $n = 1000$ with 10% noise.


https://doi.org/10.52202/079017-3899

Table 7: Average accuracy ± standard error for ablation studies of the OT and NeuralUCB components in EASE.

| Task | Noise | **OT** | **NeuralUCB** | EASE |
|---|---|---|---|---|
| LR | 0% | 50.0±0.0 | 71.7±1.7 | **80.0±2.9** |
| | 10% | 48.3±1.7 | 66.7±3.3 | **73.3±1.7** |
| | 30% | 46.7±1.7 | 66.7±1.7 | **76.7±1.7** |
| | 50% | 45.0±0.0 | 61.7±1.7 | **78.3±4.4** |
| | 70% | 38.3±3.3 | 55.0±5.0 | **66.7±1.7** |
| | 90% | 26.7±1.7 | 28.3±3.3 | **53.3±1.7** |
| LP-variant | 0% | 41.7±1.7 | **78.3±1.7** | 75.0±0.0 |
| | 10% | 40.0±0.0 | 75.0±0.0 | **80.0±2.9** |
| | 30% | 40.0±2.9 | 70.0±0.0 | **75.0±0.0** |
| | 50% | 35.0±2.9 | 71.7±1.7 | **78.3±1.7** |
| | 70% | 35.0±0.0 | 70.0±2.9 | **71.7±1.7** |
| | 90% | 50.0±0.0 | 61.7±1.7 | **66.7±1.7** |
| AG News Remap | 0% | 26.7±1.7 | 48.3±1.7 | **50.0±0.0** |
| | 10% | 15.0±0.0 | 51.7±3.3 | **51.7±1.7** |
| | 30% | 5.0±0.0 | 55.0±2.9 | **55.0±0.0** |
| | 50% | 5.0±0.0 | 41.7±3.3 | **55.0±2.9** |
| | 70% | 8.3±1.7 | 45.0±2.9 | **58.3±3.3** |
| | 90% | 5.0±0.0 | 45.0±5.8 | **53.3±1.7** |
| SST5 Reverse | 0% | 13.3±1.7 | 41.7±3.3 | **50.0±2.9** |
| | 10% | 15.0±0.0 | 48.3±4.4 | **48.3±1.7** |
| | 30% | 25.0±2.9 | 36.7±4.4 | **46.7±3.3** |
| | 50% | 15.0±0.0 | 38.3±1.7 | **46.7±3.3** |
| | 70% | 23.3±1.7 | **53.3±3.3** | 45.0±5.0 |
| | 90% | 20.0±0.0 | **33.3±3.3** | 31.7±1.7 |

Table 8: Average test accuracy ± standard error for ablation studies of the OT and NeuralUCB components in EASE.

| Task | Noise | **OT** | **NeuralUCB** | EASE |
|---|---|---|---|---|
| LR | 0% | 34.0±1.0 | 50.7±3.2 | **58.0±2.1** |
| | 10% | 29.3±0.3 | 41.0±2.9 | **42.3±0.9** |
| | 30% | 30.0±1.0 | **47.7±2.9** | 47.7±5.8 |
| | 50% | 23.0±0.6 | 49.3±1.7 | **50.7±2.2** |
| | 70% | 28.7±0.7 | 44.0±2.1 | **47.0±3.2** |
| | 90% | 32.0±0.6 | 14.7±2.2 | **39.0±7.8** |
| LP-variant | 0% | 34.7±0.7 | **56.3±4.1** | 53.0±2.3 |
| | 10% | 34.0±0.6 | **55.7±4.8** | 48.0±2.5 |
| | 30% | 36.0±1.1 | 48.7±2.6 | **50.7±1.4** |
| | 50% | 31.0±0.6 | 49.3±2.6 | **56.7±0.9** |
| | 70% | 30.3±0.9 | 50.3±3.0 | **52.0±1.0** |
| | 90% | 39.7±0.3 | 38.3±4.8 | **44.3±1.4** |
| AG News Remap | 0% | 12.3±0.7 | 29.7±0.9 | **30.3±1.2** |
| | 10% | 13.0±1.1 | **32.0±4.5** | 30.3±4.4 |
| | 30% | 4.7±0.3 | 32.0±3.1 | **40.0±1.0** |
| | 50% | 3.7±0.3 | 30.7±3.7 | **40.0±3.1** |
| | 70% | 7.0±0.0 | 34.3±1.9 | **45.0±2.0** |
| | 90% | 2.0±0.0 | 26.3±8.4 | **36.7±3.2** |
| SST5 Reverse | 0% | 9.7±0.9 | 31.3±5.2 | **34.0±3.0** |
| | 10% | 9.3±0.3 | **37.0±1.5** | 31.3±2.6 |
| | 30% | 11.7±0.7 | 21.3±7.0 | **24.3±1.8** |
| | 50% | 12.0±0.0 | 22.3±6.3 | **28.7±1.4** |
| | 70% | 14.3±0.3 | **33.7±2.2** | 33.3±3.8 |
| | 90% | 16.0±0.6 | 14.3±2.9 | **19.0±6.1** |

Table 9: Wall clock time speedups of using OT. The time in the table measures the wall clock time for each iteration of the algorithms.

| Domain size $\|Q_t\|$ | Time without OT (Sec) | Time with OT (Sec) | Speedup |
|---|---|---|---|
| 5000 | 21.48±0.12 | 5.62±0.05 | 3.8× |
| 10000 | 44.16±0.67 | 6.92±0.02 | 6.4× |
| 20000 | 91.39±1.06 | 8.82±0.12 | 10.4× |
| 50000 | 216.61±1.25 | 15.32±0.04 | 14.1× |

According to Tab. 10 and Tab. 11, EASE performs much better than the two baselines. Therefore, EASE is able to work in other general setups with a larger $k$ and $n$.

We could potentially choose more exemplars to maximize the context window, but this will incur a very high future cost of inference (i.e., the cost of actually using the exemplars for inference during production). For example, to maximize the GPT-3.5-Turbo's context window of 16385 tokens (which OpenAI charges US$0.5 per 1M tokens), each inference will cost US$0.0081925 and each run of the algorithm costs US$27.04 (assuming 165 iterations of query budget and 20 validation data samples). Similarly for GPT-4-Turbo with a context window of 128000 tokens (which OpenAI charges US$10 per 1M tokens), each inference of the algorithm now costs US$1.28 and each run of the algorithm costs US$4224. Therefore, while it is theoretically possible to use even larger $k$ which exhausts the context window, it is not economically viable to conduct experiments or adopt such an approach in practice.

Another alternative is to allow a range instead of a fixed value of $k$ to be selected. That is, if $k = 50$, we can allow the number of exemplars in the prompt to be any integer from 1 to 50. We present the additional results in Tab. 12. Firstly, EASE continues to consistently outperform the strongest baseline Best-of-N. Secondly, we also observe that the prompts with the best performance typically have a large number of exemplars, i.e., close to the max $k$ allowed. Thirdly, using $k = 50$ gives better performance than $k = 5$. Therefore, including more exemplars in the prompt usually gives a higher

Table 10: Average accuracy ± standard error for $k = 50$ and $n = 1000$.

| | **Evo** | **Best-of-N** | EASE |
|---|---|---|---|
| AG News Remap | 50.0±2.9 | 55.0±2.9 | **68.3**±1.7 |
| SST5 Reverse | 58.3±1.7 | 56.7±1.7 | **63.3**±1.7 |

Table 11: Average test accuracy ± standard error for $k = 50$ and $n = 1000$.

| | **Evo** | **Best-of-N** | EASE |
|---|---|---|---|
| AG News Remap | 12.7±3.2 | 26.3±0.9 | **38.7**±3.2 |
| SST5 Reverse | 39.0±0.6 | 38.7±1.2 | **40.0**±3.5 |

Table 12: Results for the setting that allows different number exemplars to be selected (i.e., allowing a range instead of a fixed $k$. EASE outperforms Best-of-N. We also observe high-performing prompts usually consist of more exemplars (i.e., larger average $k$).

| Tasks | $n$ | **Max** $k$ | **Best-of-N** | | EASE | |
|---|---|---|---|---|---|---|
| | | | Acc | Avg. best $k$ | Acc | Avg. best $k$ |
| LR | 100 | 5 | 60.0±2.9 | 5.0±0.0 | **76.7**±1.7 | 4.7±0.3 |
| LP-variant | 100 | 5 | 60.0±0.0 | 2.7±0.7 | **71.7**±1.7 | 4.3±0.3 |
| AG News Remap | 100 | 5 | 31.7±1.7 | 3.0±0.0 | **46.7**±1.7 | 5.0±0.0 |
| SST5 Reverse | 100 | 5 | 31.7±1.7 | 4.3±0.7 | **48.3**±1.7 | 4.7±0.3 |
| AG News Remap | 1000 | 50 | 45.0±2.9 | 25.0±0.0 | **50.0**±2.9 | 45.0±1.7 |
| SST5 Reverse | 1000 | 50 | 60.0±0.0 | 38.3±8.8 | **61.7**±1.7 | 36.0±6.7 |

performance. However, this comes at the expense of a higher query cost at test time because more tokens are used in the prompt as well.

### C.6 Ablation studies for the query budget

In the main text of the paper, we follow Lin et al. [21] and set the black-box query budget to 165 evaluations. Here, we increase the query budget to 500 iterations to study the effect of the query budget on performance. We show in Tab. 13 that increasing the budget generally improves performance.

Table 13: Average performance when the query budget is increased to 500 iterations. The performance increased as compared to Tab. 2 of the original paper. The gaps between EASE and baselines are still significant.

| Tasks (10% noise) | **Evo** | **Best-of-N** | EASE |
|---|---|---|---|
| LR | 70.0±0.0 | 70.0±0.0 | **76.7**±3.3 |
| LP-variant | 73.3±1.7 | 66.7±1.7 | **80.0**±0.0 |
| AG News Remap | 40.0±0.0 | 50.0±0.0 | **56.7**±1.7 |
| SST5 Reverse | 40.0±0.0 | 41.7±1.7 | **51.7**±1.7 |

### C.7 Ablation studies for the benefit of having an ordering-aware embedding in EASE

Note that even for the same subset of exemplars, a different ordering leads to different pre-trained embedding from embedding model $h(\cdot)$. In this section, we investigate the impact of capturing the ordering of the exemplars in the exemplar sequence $E$. To contrast the adopted approach, we propose a new embedding that simply averages the embeddings of all individual exemplars in the chosen subset. So, this embedding will be invariant with regard to the ordering of exemplars and we call it AvgEmb. As demonstrated in Tab. 14 and Tab. 15, adopting an ordering-aware embedding in EASE results in better overall performance as compared to AvgEmd which disregards exemplar ordering.

Notably, AvgEmd presents a trade-off between efficiency and effectiveness. AvgEmd is advantageous in terms of efficiency as it eliminates the on-the-fly embedding entirely. However, there is no free lunch: Such computational simplification comes at the cost of performance due to the loss of order information. Though not as good, note that AvgEmb still achieves a decent and competitive performance. The practitioners can balance the trade-off and select the most suitable method.


https://doi.org/10.52202/079017-3899

Table 14: Average accuracy ± standard error for ablation studies of using an embedding without considering order.

| Task | Noise | **AvgEmb** | EASE |
|---|---|---|---|
| LR | 0% | 76.7±1.7 | **80.0**±2.9 |
| | 10% | 73.3±4.4 | **73.3**±1.7 |
| | 30% | 70.0±0.0 | **76.7**±1.7 |
| | 50% | 68.3±4.4 | **78.3**±4.4 |
| | 70% | **66.7**±1.7 | **66.7**±1.7 |
| | 90% | 51.7±1.7 | **53.3**±1.7 |
| LP-variant | 0% | **76.7**±1.7 | 75.0±0.0 |
| | 10% | 75.0±0.0 | **80.0**±2.9 |
| | 30% | 70.0±0.0 | **75.0**±0.0 |
| | 50% | **78.3**±1.7 | **78.3**±1.7 |
| | 70% | **75.0**±2.9 | 71.7±1.7 |
| | 90% | 61.7±1.7 | **66.7**±1.7 |
| AG News Remap | 0% | **53.3**±1.7 | 50.0±0.0 |
| | 10% | 50.0±5.0 | **51.7**±1.7 |
| | 30% | 51.7±1.7 | **55.0**±0.0 |
| | 50% | 50.0±0.0 | **55.0**±2.9 |
| | 70% | **58.3**±1.7 | 58.3±3.3 |
| | 90% | 53.3±6.0 | **53.3**±1.7 |
| SST5 Reverse | 0% | 46.7±6.7 | **50.0**±2.9 |
| | 10% | **50.0**±2.9 | 48.3±1.7 |
| | 30% | 45.0±2.9 | **46.7**±3.3 |
| | 50% | **48.3**±1.7 | 46.7±3.3 |
| | 70% | 41.7±4.4 | **45.0**±5.0 |
| | 90% | **31.7**±1.7 | **31.7**±1.7 |
| # best-performing tasks | | 9 | 18 |

Table 15: Average test accuracy ± standard error for ablation studies of using an embedding without considering order.

| Task | Noise | **AvgEmb** | EASE |
|---|---|---|---|
| LR | 0% | 48.7±1.2 | **58.0**±2.1 |
| | 10% | **48.7**±2.9 | 42.3±0.9 |
| | 30% | **53.0**±4.9 | 47.7±5.8 |
| | 50% | 46.0±5.0 | **50.7**±2.2 |
| | 70% | 44.3±6.4 | **47.0**±3.2 |
| | 90% | 28.7±2.7 | **39.0**±7.8 |
| LP-variant | 0% | 47.7±1.9 | **53.0**±2.3 |
| | 10% | **48.0**±1.7 | 48.0±2.5 |
| | 30% | 49.3±1.8 | **50.7**±1.4 |
| | 50% | 48.0±1.5 | **56.7**±0.9 |
| | 70% | **54.3**±1.3 | 52.0±1.0 |
| | 90% | 38.3±0.9 | **44.3**±1.4 |
| AG News Remap | 0% | **33.3**±2.3 | 30.3±1.2 |
| | 10% | **32.0**±0.6 | 30.3±4.4 |
| | 30% | 29.7±1.2 | **40.0**±1.0 |
| | 50% | 28.7±1.4 | **40.0**±3.1 |
| | 70% | 40.0±5.0 | **45.0**±2.0 |
| | 90% | 36.3±4.1 | **36.7**±3.2 |
| SST5 Reverse | 0% | 31.0±4.6 | **34.0**±3.0 |
| | 10% | 31.3±3.0 | **31.3**±2.6 |
| | 30% | **27.7**±1.4 | 24.3±1.8 |
| | 50% | **33.7**±1.4 | 28.7±1.4 |
| | 70% | 25.7±7.2 | **33.3**±3.8 |
| | 90% | 13.7±0.3 | **19.0**±6.1 |
| # best-performing tasks | | 8 | 16 |

Table 16: Average accuracy ± standard error when using our EASE to select exemplars for different target black-box models.

| Task (with 10% noise) | **GPT-4-V** | **GPT-4-Turbo** | **Gemini Pro** |
|---|---|---|---|
| LR | 1.7±1.7 | 3.3±1.7 | 83.3±4.4 |
| LP-variant | 90.0±0.0 | 90.0±0.0 | 31.7±1.7 |
| AG News Remap | 50.0±0.0 | 35.0±2.9 | 53.3±4.4 |
| SST5 Reverse | 21.7±1.7 | 41.7±1.7 | 36.7±1.7 |

Table 17: Average test accuracy ± standard error when using our EASE to select exemplars for different target black-box models.

| Task (with 10% noise) | **GPT-4-V** | **GPT-4-Turbo** | **Gemini Pro** |
|---|---|---|---|
| LR | 0.0±0.0 | 0.0±0.0 | 51.3±3.5 |
| LP-variant | 83.7±1.3 | 77.7±1.9 | 14.3±1.4 |
| AG News Remap | 25.7±1.9 | 25.7±3.0 | 32.7±9.4 |
| SST5 Reverse | 13.3±1.4 | 28.3±3.8 | 24.0±3.0 |

### C.8 Ablation studies for optimizing exemplars for different black-box LLMs in EASE

We perform exemplar selection for other target black-box models that are not GPT-3.5 which we used for all other experiments previously. We use GPT-4-V (i.e., "gpt-4-turbo-2024-04-09" with vision capability), GPT-4-Turbo (i.e., 'gpt-4-1106-preview' without vision capability) and Gemini Pro [41] (i.e., "gemini-1.0-pro") for our experiments. Tab. 16 and Tab. 17 show that our EASE is able to select effective exemplars for different black-box models. Note that the performance of GPT-4-V and GPT-4-Turbo is comparable to or worse than that of GPT-3.5 in some tasks. This may be attributed to significant variations in performance across different versions (i.e., checkpoints) of the GPT models. For example, existing work by Chen et al. [6] has shown that GPT-4's mathematical ability varies a lot across different checkpoints, with some exhibiting poor performance on mathematical problems. This variability partially explains the suboptimal performance of GPT-4-V and GPT-4-Turbo on the task of LR. Additionally, it is worth investigating, in future work, the significance of exemplar selection as the LLMs continue to become increasingly powerful.

### C.9 Ablation studies for using different embedding models in EASE

For our main experiments, we use MPNet as the embedding model for our EASE. Here, we use MiniLM [43] and CLIP [35] to obtain the embedding for our EASE, respectively. Tab. 18 and Tab. 19

Table 18: Average accuracy ± standard error when using different embedding models for our EASE.

| Task (with 10% noise) | **MiniLM** | **CLIP** |
|---|---|---|
| LR | 73.3±1.7 | 68.3±1.7 |
| LP-variant | 76.7±1.7 | 70.0±2.9 |
| AG News Remap | 43.3±1.7 | 46.7±3.3 |
| SST5 Reverse | 43.3±1.7 | 45.0±0.0 |

Table 19: Average test accuracy ± standard error when using different embedding models for our EASE.

| Task (with 10% noise) | **MiniLM** | **CLIP** |
|---|---|---|
| LR | 51.3±1.3 | 50.3±5.5 |
| LP-variant | 56.3±0.9 | 49.7±1.4 |
| AG News Remap | 29.3±2.7 | 29.3±0.3 |
| SST5 Reverse | 29.7±6.4 | 33.3±4.8 |

Table 20: Average accuracy ± standard error for ablation studies of the exploration parameter $\nu$.

| Task | Noise | $\nu = 0$ | $\nu = 0.01$ | $\nu = 0.1$ | $\nu = 1$ |
|---|---|---|---|---|---|
| LP-variant | 0% | **75.0**±0.0 | **75.0**±0.0 | 71.7±1.7 | 70.0±0.0 |
| | 10% | 75.0±2.9 | **80.0**±2.9 | 71.7±1.7 | 73.3±1.7 |
| | 30% | 73.3±1.7 | **75.0**±0.0 | 68.3±1.7 | 71.7±1.7 |
| | 50% | 76.7±3.3 | **78.3**±1.7 | 75.0±2.9 | 71.7±1.7 |
| | 70% | **75.0**±0.0 | 71.7±1.7 | 73.3±1.7 | 70.0±0.0 |
| | 90% | 63.3±1.7 | **66.7**±1.7 | 63.3±1.7 | 63.3±1.7 |

Table 21: Average test accuracy ± standard error for ablation studies of the exploration parameter $\nu$.

| Task | Noise | $\nu = 0$ | $\nu = 0.01$ | $\nu = 0.1$ | $\nu = 1$ |
|---|---|---|---|---|---|
| LP-variant | 0% | 49.3±2.4 | **53.0**±2.3 | 52.0±1.1 | 50.0±3.5 |
| | 10% | 46.0±1.0 | 48.0±2.5 | **51.3**±2.2 | 49.3±1.9 |
| | 30% | 49.3±1.3 | 50.7±1.4 | 43.7±3.8 | **52.3**±2.4 |
| | 50% | 54.3±2.6 | **56.7**±0.9 | 50.3±2.7 | 47.0±1.1 |
| | 70% | 49.0±4.0 | **52.0**±1.0 | 46.0±2.5 | 47.0±1.1 |
| | 90% | 41.3±2.3 | **44.3**±1.4 | 36.7±2.6 | 40.3±1.4 |

show that our EASE achieves competitive performance using different embedding models. Therefore, our EASE is general in the sense that different embedding models can be used. It might be worth investigating as a future direction to develop or finetune embedding models specifically for the purpose of prompt optimization, such that it captures important aspects of the prompt in the latent space.

### C.10 Ablation studies for the degree of exploration

One important hyperparameter of the NeuralUCB algorithm utilized in the paper is $\nu$ (see (3)), which controls the degree of exploration performed during the optimization process. Our finding indicates that a small $\nu$ in EASE results in better exemplar selection performance. As demonstrated in Tab. 20 and Tab. 21, having a near-zero (i.e., $\nu = 0.01$) exploration is the best. This may be attributed to our sub-sampling of the domain space $\Omega$, which is motivated by practically reducing computational complexities, and at the same time inherently serves as a form of exploration. Given the enormous size of the $\Omega$, sub-sampling to $Q_t$ likely generates a different exemplar sequences space within each iteration of the algorithm. This observation is further supported by Bastani et al. [2], who found that a greedy algorithm can be rate optimal given sufficient randomness in the observed contexts of the bandits algorithm. Therefore, only a small extent of exploration is required for EASE.

To further support the above observation empirically, we conduct additional experiments with the domain space $Q_t$ fixed throughout. When $|Q_t| = 1000$ and fixed, we observe results in Tab. 22 and Tab. 23 that a larger value of $\nu$ performs better. Thus, exploration is essential when there is no randomness in the context (i.e., the domain space $Q_t$). However, given sufficient randomness in our EASE algorithm from sub-sampling, we adopt a small $\nu = 0.01$ for optimal performance.

Table 22: Average accuracy ± standard error for ablation studies of exploration in a fixed domain.

| Task | Noise | $\nu = 0$ | $\nu = 0.01$ | $\nu = 0.1$ | $\nu = 1$ |
|---|---|---|---|---|---|
| LP-variant | 0% | 60.0±0.0 | 66.7±1.7 | **68.3**±1.7 | 66.7±1.7 |
| | 10% | 60.0±0.0 | 61.7±1.7 | **66.7**±1.7 | 65.0±2.9 |
| | 30% | 58.3±1.7 | 58.3±1.7 | **65.0**±2.9 | **65.0**±2.9 |
| | 50% | 60.0±2.9 | 65.0±0.0 | 61.7±1.7 | **66.7**±1.7 |
| | 70% | 50.0±0.0 | 51.7±1.7 | 58.3±1.7 | **61.7**±3.3 |
| | 90% | 36.7±1.7 | 36.7±1.7 | 45.0±0.0 | **48.3**±1.7 |

Table 23: Average test accuracy ± standard error for ablation studies of the exploration in a fixed domain.

| Task | Noise | $\nu = 0$ | $\nu = 0.01$ | $\nu = 0.1$ | $\nu = 1$ |
|---|---|---|---|---|---|
| LP-variant | 0% | 38.3±0.9 | 51.0±3.2 | **53.7**±3.3 | 45.0±1.1 |
| | 10% | 41.7±2.7 | 42.7±1.9 | 47.0±1.5 | **49.0**±2.1 |
| | 30% | 39.3±2.3 | 40.3±3.3 | **43.3**±1.8 | 43.0±1.1 |
| | 50% | 43.3±2.3 | 47.7±0.9 | 41.7±2.3 | **48.7**±2.4 |
| | 70% | 39.7±1.4 | 42.0±3.1 | 39.7±1.9 | **48.7**±0.3 |
| | 90% | 20.7±0.3 | 22.3±0.9 | 29.0±3.6 | **29.0**±1.1 |


https://doi.org/10.52202/079017-3899

Table 24: Average accuracy ± standard error when asking GPT to directly output the best exemplars.

| Task (with 10% noise) | GPT Select | EASE |
|---|---|---|
| LR | 35.0±7.6 | **73.3**±1.7 |
| LP-variant | 58.3±3.3 | **80.0**±2.9 |
| AG News Remap | 10.0±0.0 | **51.7**±1.7 |
| SST5 Reverse | 10.0±0.0 | **48.3**±1.7 |

Table 25: Average test accuracy ± standard error when asking GPT to directly output the best exemplars.

| Task (with 10% noise) | GPT Select | EASE |
|---|---|---|
| LR | 29.0±9.0 | **42.3**±0.9 |
| LP-variant | 44.0±1.0 | **48.0**±2.5 |
| AG News Remap | 12.7±0.3 | **30.3**±4.4 |
| SST5 Reverse | 10.0±0.0 | **31.3**±2.6 |

Table 26: Average performance for tasks involving reasoning chains.

| Tasks | Evo | Best-of-N | EASE |
|---|---|---|---|
| MATH [15] | 61.7±1.7 | 60.0±0.0 | **65.0**±0.0 |
| GSM8K [7] | 70.0±2.9 | 68.3±1.7 | **75.0**±0.0 |
| AQuA-RAT [22] | 46.7±1.7 | 43.3±1.7 | **48.3**±1.7 |

### C.11 Ablation studies: Asking GPT to directly select exemplars

One may wonder whether ChatGPT can directly help us select the exemplars for in-context learning. To test the possibility of directly utilizing ChatGPT, we use the following prompt to query the GPT:

> *"You are asked to perform a task that is described by the examples below, the goal is to correctly give output based on the input. Given the numbered list of examples below, pick 5 of them that will best serve as examples for in-context learning for this task. Only output the list of numbers."*

We provide all the exemplars in $D$ in the form of a numbered list to GPT together with the prompt above, and obtain the GPT-selected exemplars. We call this method GPT Select. As shown in Tab. 24 and Tab. 25, the exemplars directly selected by GPT perform worse than EASE for ICL.

### C.12 Additional experiments on tasks with reasoning chains or open-ended generation

Note that the tasks included in the main experiments of the paper already include open-ended generation tasks. Tasks like Auto Categorization, Word Sorting, LR, and LP-variant are beyond simple direct classification-type labels. For example, Auto Categorization requires outputting an open-ended sentence that categorizes the inputs well; LP-variant does open-ended sentence translation following a set of nontrivial rules. Therefore, our EASE demonstrates effectiveness in challenging open-ended generation tasks.

We conduct additional experiments for tasks with reasoning chains, including MATH [15], GSM8K [7], and AQuA-RAT [22]. These tasks are typically harder and involve longer responses. From Tab. 26, EASE works well for these tasks and the advantages are significant.

### C.13 Additional experiments on more datasets

We conduct experiments on a number of additional real benchmarks. We present results on the benchmarks used in the TEMPERA [49] paper in Tab. 27, including MR, Yelp P., CR, MNLI & MRPC. Note that these tasks are overly simple to distinguish the effectiveness of EASE because all of them achieve above 80% accuracy and are hard to improve further using mere in-context examples.

Therefore, we refer the reviewer to more complicated benchmarks in the main paper. Additionally, real tasks in Sec. C.12 also demonstrated good performance of EASE on MATH, GSM8K & AQuA-RAT with Chain-of-Thought reasoning, making the baseline comparisons in our paper more convincing.

Table 27: Average performance on benchmarks used in TEMPERA [49].

| Tasks | **Evo** | **Best-of-N** | EASE |
|---|---|---|---|
| MR | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ |
| Yelp P. | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ | **95.0**$_{\pm 0.0}$ |
| CR | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| MNLI | **81.7**$_{\pm 1.7}$ | 80.0$_{\pm 0.0}$ | **81.7**$_{\pm 1.7}$ |
| MRPC | **81.7**$_{\pm 1.7}$ | **81.7**$_{\pm 1.7}$ | **81.7**$_{\pm 1.7}$ |

Table 28: Average performance on Meta's newest open-weight model Llama-3.1-8B-Instruct.

| Tasks (with 50% noise) | **Evo** | **Best-of-N** | EASE |
|---|---|---|---|
| LR | 11.7$_{\pm 1.7}$ | 15.0$_{\pm 2.9}$ | **35.0**$_{\pm 2.9}$ |
| LP-variant | 11.7$_{\pm 1.7}$ | 15.0$_{\pm 0.0}$ | **25.0**$_{\pm 0.0}$ |
| AG News Remap | 31.7$_{\pm 1.7}$ | 31.7$_{\pm 1.7}$ | **33.3**$_{\pm 1.7}$ |
| SST5 Reverse | **31.7**$_{\pm 3.3}$ | 30.0$_{\pm 0.0}$ | 28.3$_{\pm 3.3}$ |

### C.14 Open-weight models for reproducibility

For reproducibility's sake, we conduct additional experiments with open-weight models. We use a representative (Meta's newest) model Llama-3.1-8B-Instruct as the target model. We present the results in Tab. 28. Importantly, the results are consistent with the original conclusions we drew for the black-box models in the paper. The model is also much smaller than GPT-3.5, hence making it easier for others to deploy and reproduce the results.

### C.15 Test accuracies for all tables

In Tab. 29, Tab. 30, Tab. 31 and Tab. 32, we present the test accuracies for all tables presented in the main paper. Note that we report validation accuracies in the main tables because the effectiveness of the optimization strategy is directly reflected by the maximized value of the objective function in (1) at the end of all iterations. Nevertheless, we show all test accuracies here for reference and completeness. The test accuracy can be affected by the difference in the distribution of the validation and test data, which is out of our control. This difference in distribution is significant in our case because the validation set only contains 20 data points (limited by the fact that querying the GPT-3.5 API is expensive). This setup is disadvantageous for EASE because the optimized exemplar is "overfitted" to the validation set. This test performance is expected to improve much if we use a larger validation set (e.g., with 100 validation samples).

## D Other discussions

### D.1 Further validations of "out-of-distribution" tasks benchmarks

In Sec. 4.3, we propose new families of "out-of-distribution" tasks that highlight the importance of high-quality exemplars. We note that "out-of-distribution" tasks are defined loosely here as tasks which the LLM are not already well trained.

However, we cannot confirm whether a task is already well-trained without access to the training data. We instead perform an empirical validation to show that "out-of-distribution" tasks are less likely to be well-trained, and hence suitable dataset benchmarks to access the performance of algorithms. As shown in Tab. 33 and Tab. 34, we performed random exemplar in-context prompting and discovered that it achieved an average performance of 17.6% for "out-of-distribution" tasks, which contrasts with the average performance of 64.7% for Instruction Induction benchmark tasks. This demonstrates that the model is likely to have less knowledge about our defined "out-of-distribution" tasks. Additionally, we can also see that the performance gain of EASE is higher for "out-of-distribution" tasks.


https://doi.org/10.52202/079017-3899

Table 29: Test accuracies counterpart of Tab. 1 over 3 independent trials.

| | DPP | MMD | OT | Cosine | BM25 | Active | Inf | Evo | Best-of-N | EASE |
|---|---|---|---|---|---|---|---|---|---|---|
| active_to_passive | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| antonyms | 75.7$_{\pm 0.3}$ | **86.0**$_{\pm 0.0}$ | 85.3$_{\pm 0.7}$ | 77.7$_{\pm 0.3}$ | 80.7$_{\pm 0.3}$ | 80.0$_{\pm 2.5}$ | 82.3$_{\pm 0.3}$ | 82.0$_{\pm 0.0}$ | 84.3$_{\pm 0.3}$ | 84.7$_{\pm 0.3}$ |
| auto_categorization | 28.7$_{\pm 0.3}$ | 25.7$_{\pm 0.7}$ | 24.0$_{\pm 0.6}$ | **39.3**$_{\pm 0.9}$ | 27.0$_{\pm 1.0}$ | 32.7$_{\pm 1.0}$ | 35.0$_{\pm 1.5}$ | 17.3$_{\pm 0.3}$ | 30.7$_{\pm 0.7}$ | 34.0$_{\pm 2.1}$ |
| diff | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| first_word_letter | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| larger_animal | 69.0$_{\pm 0.6}$ | 78.7$_{\pm 1.2}$ | 83.7$_{\pm 0.7}$ | 77.7$_{\pm 0.7}$ | 84.0$_{\pm 1.0}$ | 61.0$_{\pm 1.7}$ | 84.7$_{\pm 0.3}$ | **90.0**$_{\pm 0.6}$ | 87.7$_{\pm 2.9}$ | 88.0$_{\pm 0.0}$ |
| letters_list | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| negation | 85.7$_{\pm 0.7}$ | 87.3$_{\pm 0.7}$ | 86.3$_{\pm 0.3}$ | 87.7$_{\pm 0.3}$ | 83.3$_{\pm 0.3}$ | 85.7$_{\pm 1.1}$ | 83.7$_{\pm 0.3}$ | 86.3$_{\pm 0.3}$ | 85.0$_{\pm 0.6}$ | **89.0**$_{\pm 0.6}$ |
| num_to_verbal | 97.3$_{\pm 0.3}$ | 98.7$_{\pm 0.3}$ | **99.0**$_{\pm 0.0}$ | **99.0**$_{\pm 0.0}$ | 96.0$_{\pm 0.0}$ | 97.7$_{\pm 0.7}$ | 98.0$_{\pm 0.0}$ | 96.7$_{\pm 0.3}$ | 96.7$_{\pm 0.3}$ | 96.3$_{\pm 0.3}$ |
| object_counting | 45.3$_{\pm 0.9}$ | 48.3$_{\pm 0.9}$ | 40.7$_{\pm 0.3}$ | 27.0$_{\pm 2.5}$ | 31.3$_{\pm 3.8}$ | 48.0$_{\pm 2.0}$ | **58.0**$_{\pm 0.6}$ | 42.3$_{\pm 0.3}$ | 52.3$_{\pm 3.3}$ | 52.0$_{\pm 2.1}$ |
| orthography_starts_with | 30.3$_{\pm 0.9}$ | 42.0$_{\pm 0.6}$ | 65.7$_{\pm 0.3}$ | 65.7$_{\pm 0.9}$ | **71.0**$_{\pm 0.6}$ | 40.3$_{\pm 7.8}$ | 69.3$_{\pm 0.9}$ | 47.0$_{\pm 1.0}$ | 63.3$_{\pm 2.4}$ | 69.3$_{\pm 0.7}$ |
| rhymes | 58.0$_{\pm 1.0}$ | 42.0$_{\pm 0.6}$ | 11.3$_{\pm 0.9}$ | **99.0**$_{\pm 0.0}$ | 64.3$_{\pm 1.9}$ | 47.3$_{\pm 6.9}$ | 59.3$_{\pm 15.4}$ | 98.3$_{\pm 0.3}$ | 96.0$_{\pm 0.0}$ | 97.7$_{\pm 1.3}$ |
| second_word_letter | 17.3$_{\pm 0.9}$ | 31.0$_{\pm 0.0}$ | 32.3$_{\pm 0.7}$ | 42.0$_{\pm 3.1}$ | 42.3$_{\pm 0.9}$ | 27.0$_{\pm 8.2}$ | 45.7$_{\pm 1.2}$ | 40.3$_{\pm 0.3}$ | 38.7$_{\pm 0.3}$ | **48.3**$_{\pm 1.4}$ |
| sentence_similarity | 19.0$_{\pm 0.6}$ | 13.7$_{\pm 0.3}$ | 38.3$_{\pm 1.4}$ | **41.0**$_{\pm 0.6}$ | 33.7$_{\pm 2.3}$ | 19.0$_{\pm 1.7}$ | 20.0$_{\pm 5.0}$ | 18.0$_{\pm 0.0}$ | 31.3$_{\pm 2.7}$ | 32.7$_{\pm 1.2}$ |
| sentiment | 91.3$_{\pm 0.3}$ | 89.0$_{\pm 0.0}$ | 87.0$_{\pm 0.0}$ | 90.3$_{\pm 0.3}$ | 90.7$_{\pm 0.7}$ | **91.7**$_{\pm 0.7}$ | 89.0$_{\pm 0.0}$ | 89.7$_{\pm 0.3}$ | 89.3$_{\pm 0.3}$ | 89.0$_{\pm 0.0}$ |
| singular_to_plural | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| sum | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 2.0$_{\pm 0.0}$ | 34.7$_{\pm 26.7}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ |
| synonyms | 14.0$_{\pm 0.6}$ | 14.3$_{\pm 0.7}$ | 11.0$_{\pm 0.0}$ | 14.0$_{\pm 0.0}$ | 13.7$_{\pm 0.3}$ | 13.0$_{\pm 0.5}$ | 14.7$_{\pm 0.9}$ | 13.3$_{\pm 0.9}$ | **18.3**$_{\pm 0.3}$ | 13.7$_{\pm 0.3}$ |
| taxonomy_animal | 47.7$_{\pm 0.3}$ | 44.7$_{\pm 1.2}$ | 77.3$_{\pm 1.9}$ | 73.3$_{\pm 0.3}$ | 67.3$_{\pm 5.9}$ | 46.3$_{\pm 7.1}$ | 62.3$_{\pm 8.9}$ | 32.7$_{\pm 0.7}$ | 73.7$_{\pm 4.2}$ | **78.7**$_{\pm 1.9}$ |
| translation_en-de | 83.7$_{\pm 0.3}$ | 84.3$_{\pm 0.3}$ | 83.3$_{\pm 0.3}$ | 83.0$_{\pm 0.6}$ | 83.3$_{\pm 1.3}$ | 80.3$_{\pm 1.9}$ | **85.0**$_{\pm 1.1}$ | 83.3$_{\pm 0.3}$ | 82.3$_{\pm 0.3}$ | 83.3$_{\pm 0.3}$ |
| translation_en-es | 83.7$_{\pm 0.3}$ | **89.7**$_{\pm 0.3}$ | 88.3$_{\pm 0.3}$ | 87.3$_{\pm 1.2}$ | 87.3$_{\pm 1.2}$ | 88.0$_{\pm 0.8}$ | 86.7$_{\pm 1.4}$ | 84.3$_{\pm 0.3}$ | 84.3$_{\pm 0.3}$ | 84.7$_{\pm 0.9}$ |
| translation_en-fr | 83.0$_{\pm 0.0}$ | 87.7$_{\pm 0.3}$ | 87.3$_{\pm 0.3}$ | 87.0$_{\pm 0.0}$ | 87.0$_{\pm 0.0}$ | 83.3$_{\pm 1.7}$ | 87.7$_{\pm 0.7}$ | 86.7$_{\pm 0.3}$ | **88.3**$_{\pm 0.3}$ | 85.0$_{\pm 1.0}$ |
| word_sorting | 8.7$_{\pm 0.7}$ | 65.0$_{\pm 0.6}$ | 66.7$_{\pm 1.2}$ | 66.0$_{\pm 1.1}$ | 43.7$_{\pm 2.3}$ | **71.0**$_{\pm 1.4}$ | 69.3$_{\pm 1.7}$ | 61.3$_{\pm 1.2}$ | 63.3$_{\pm 0.7}$ | 69.7$_{\pm 0.3}$ |
| word_unscrambling | 61.3$_{\pm 0.3}$ | 58.7$_{\pm 0.7}$ | **63.3**$_{\pm 0.3}$ | 57.0$_{\pm 1.1}$ | 61.3$_{\pm 0.7}$ | 60.7$_{\pm 1.7}$ | 58.3$_{\pm 1.4}$ | 53.7$_{\pm 0.3}$ | 60.3$_{\pm 0.3}$ | 62.0$_{\pm 2.1}$ |
| # best-performing tasks | 4 | 6 | 6 | 8 | 5 | 7 | 8 | 7 | 8 | 9 |

Table 30: Test accuracies counterpart of Tab. 2 over 3 independent trials.

| Type | Task | Noise | DPP | MMD | OT | Cosine | BM25 | Active | Inf | Evo | Best-of-N | EASE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rule-based tasks | LR | 0% | 36.7$_{\pm 0.7}$ | 38.0$_{\pm 1.0}$ | 34.0$_{\pm 1.0}$ | 48.0$_{\pm 1.1}$ | 40.7$_{\pm 2.3}$ | 32.0$_{\pm 6.6}$ | 43.3$_{\pm 5.8}$ | 38.0$_{\pm 0.6}$ | 47.7$_{\pm 0.3}$ | **58.0**$_{\pm 2.1}$ |
| | | 10% | 7.3$_{\pm 0.9}$ | 33.3$_{\pm 0.7}$ | 29.3$_{\pm 0.3}$ | 40.3$_{\pm 3.8}$ | 41.0$_{\pm 2.9}$ | 0.0$_{\pm 0.0}$ | 46.3$_{\pm 1.2}$ | 38.3$_{\pm 0.3}$ | **47.7**$_{\pm 0.7}$ | 42.3$_{\pm 0.9}$ |
| | | 30% | 12.0$_{\pm 0.6}$ | 23.3$_{\pm 0.3}$ | 30.0$_{\pm 1.0}$ | 45.7$_{\pm 4.8}$ | 29.7$_{\pm 2.9}$ | 9.0$_{\pm 5.8}$ | 35.0$_{\pm 4.7}$ | 38.3$_{\pm 1.2}$ | 43.0$_{\pm 0.6}$ | **47.7**$_{\pm 5.8}$ |
| | | 50% | 3.0$_{\pm 1.0}$ | 33.7$_{\pm 0.3}$ | 23.0$_{\pm 0.6}$ | 43.7$_{\pm 2.3}$ | 33.3$_{\pm 0.7}$ | 0.0$_{\pm 0.0}$ | 47.0$_{\pm 2.1}$ | 32.7$_{\pm 0.9}$ | 30.3$_{\pm 1.8}$ | **50.7**$_{\pm 2.2}$ |
| | | 70% | 0.0$_{\pm 0.0}$ | 38.7$_{\pm 0.3}$ | 28.7$_{\pm 0.7}$ | 44.0$_{\pm 3.5}$ | 34.0$_{\pm 1.1}$ | 1.7$_{\pm 1.4}$ | 31.7$_{\pm 3.8}$ | 30.3$_{\pm 0.7}$ | 26.3$_{\pm 5.3}$ | **47.0**$_{\pm 3.2}$ |
| | | 90% | 0.0$_{\pm 0.0}$ | 32.7$_{\pm 1.3}$ | 32.0$_{\pm 0.6}$ | 37.3$_{\pm 4.8}$ | 2.3$_{\pm 0.9}$ | 0.0$_{\pm 0.0}$ | 7.7$_{\pm 1.9}$ | 5.0$_{\pm 0.6}$ | 14.0$_{\pm 0.6}$ | **39.0**$_{\pm 7.8}$ |
| | LP-variant | 0% | 39.3$_{\pm 0.7}$ | 35.0$_{\pm 0.6}$ | 34.7$_{\pm 0.7}$ | 47.3$_{\pm 0.3}$ | 36.3$_{\pm 1.4}$ | 34.0$_{\pm 3.7}$ | **53.3**$_{\pm 1.3}$ | 38.3$_{\pm 0.9}$ | 51.7$_{\pm 3.3}$ | 53.0$_{\pm 2.3}$ |
| | | 10% | 0.0$_{\pm 0.0}$ | 37.0$_{\pm 0.6}$ | 34.0$_{\pm 0.6}$ | 49.0$_{\pm 1.1}$ | 39.3$_{\pm 2.2}$ | 39.0$_{\pm 0.8}$ | **52.3**$_{\pm 2.6}$ | 37.7$_{\pm 0.7}$ | 48.0$_{\pm 1.1}$ | 48.0$_{\pm 2.5}$ |
| | | 30% | 0.0$_{\pm 0.0}$ | 44.3$_{\pm 0.7}$ | 36.0$_{\pm 1.1}$ | 38.7$_{\pm 4.3}$ | 34.3$_{\pm 1.9}$ | 36.7$_{\pm 2.4}$ | 48.0$_{\pm 3.1}$ | 35.7$_{\pm 0.3}$ | 49.3$_{\pm 0.7}$ | **50.7**$_{\pm 1.4}$ |
| | | 50% | 0.0$_{\pm 0.0}$ | 53.3$_{\pm 1.2}$ | 31.0$_{\pm 0.6}$ | 47.0$_{\pm 0.6}$ | 34.3$_{\pm 1.3}$ | 12.7$_{\pm 5.2}$ | 42.7$_{\pm 2.0}$ | 34.0$_{\pm 0.6}$ | 49.3$_{\pm 0.3}$ | **56.7**$_{\pm 0.9}$ |
| | | 70% | 0.0$_{\pm 0.0}$ | 39.0$_{\pm 1.5}$ | 30.3$_{\pm 0.9}$ | 46.7$_{\pm 1.3}$ | 36.3$_{\pm 0.7}$ | 4.3$_{\pm 3.5}$ | 51.0$_{\pm 1.7}$ | 33.3$_{\pm 0.7}$ | 31.3$_{\pm 0.9}$ | **52.0**$_{\pm 1.0}$ |
| | | 90% | 0.0$_{\pm 0.0}$ | 35.0$_{\pm 0.6}$ | 39.7$_{\pm 0.3}$ | 35.3$_{\pm 1.9}$ | 0.0$_{\pm 0.0}$ | 0.0$_{\pm 0.0}$ | 41.0$_{\pm 1.7}$ | 16.7$_{\pm 0.9}$ | 28.0$_{\pm 0.6}$ | **44.3**$_{\pm 1.4}$ |
| Re-mapped label tasks | AG News Remap | 0% | 7.0$_{\pm 0.6}$ | 7.0$_{\pm 0.0}$ | 12.3$_{\pm 0.7}$ | 30.3$_{\pm 1.4}$ | 28.3$_{\pm 1.3}$ | 4.3$_{\pm 1.1}$ | 9.0$_{\pm 2.5}$ | 11.3$_{\pm 0.3}$ | 19.3$_{\pm 0.7}$ | **30.3**$_{\pm 1.2}$ |
| | | 10% | 2.0$_{\pm 0.0}$ | 7.0$_{\pm 0.0}$ | 13.0$_{\pm 1.1}$ | 17.0$_{\pm 4.0}$ | 19.0$_{\pm 4.5}$ | 6.0$_{\pm 0.8}$ | 13.0$_{\pm 3.2}$ | 12.0$_{\pm 0.6}$ | 30.0$_{\pm 0.6}$ | **30.3**$_{\pm 4.4}$ |
| | | 30% | 3.7$_{\pm 0.3}$ | 1.0$_{\pm 0.0}$ | 4.7$_{\pm 0.3}$ | 28.0$_{\pm 2.0}$ | 22.3$_{\pm 0.7}$ | 7.7$_{\pm 2.2}$ | 5.0$_{\pm 0.6}$ | 12.3$_{\pm 0.3}$ | 28.0$_{\pm 5.0}$ | **40.0**$_{\pm 1.0}$ |
| | | 50% | 4.7$_{\pm 0.3}$ | 3.0$_{\pm 0.0}$ | 3.7$_{\pm 0.3}$ | 27.3$_{\pm 2.3}$ | 21.3$_{\pm 2.7}$ | 8.7$_{\pm 3.1}$ | 13.0$_{\pm 3.5}$ | 6.7$_{\pm 0.3}$ | 21.7$_{\pm 2.3}$ | **40.0**$_{\pm 3.1}$ |
| | | 70% | 2.0$_{\pm 0.0}$ | 23.3$_{\pm 0.3}$ | 7.0$_{\pm 0.0}$ | 20.0$_{\pm 1.0}$ | 16.0$_{\pm 0.6}$ | 15.7$_{\pm 7.1}$ | 5.0$_{\pm 1.1}$ | 4.7$_{\pm 0.3}$ | 36.0$_{\pm 0.6}$ | **45.0**$_{\pm 2.0}$ |
| | | 90% | 8.0$_{\pm 0.6}$ | 21.0$_{\pm 1.1}$ | 2.0$_{\pm 0.0}$ | 22.7$_{\pm 7.0}$ | 2.0$_{\pm 0.0}$ | 11.7$_{\pm 1.9}$ | 23.0$_{\pm 1.5}$ | 6.3$_{\pm 0.3}$ | 27.3$_{\pm 1.9}$ | **36.7**$_{\pm 3.2}$ |
| | SST5 Reverse | 0% | 13.3$_{\pm 0.7}$ | 11.7$_{\pm 0.7}$ | 9.7$_{\pm 0.9}$ | 22.0$_{\pm 3.6}$ | 18.3$_{\pm 0.7}$ | 10.3$_{\pm 0.3}$ | 21.7$_{\pm 5.5}$ | 11.3$_{\pm 0.3}$ | 23.7$_{\pm 0.7}$ | **34.0**$_{\pm 3.0}$ |
| | | 10% | 13.3$_{\pm 0.9}$ | 11.3$_{\pm 0.3}$ | 9.3$_{\pm 0.3}$ | 20.0$_{\pm 2.6}$ | 18.0$_{\pm 1.0}$ | 11.7$_{\pm 1.5}$ | 27.0$_{\pm 1.1}$ | 12.0$_{\pm 0.6}$ | 23.0$_{\pm 0.6}$ | **31.3**$_{\pm 2.6}$ |
| | | 30% | 15.7$_{\pm 0.3}$ | 13.3$_{\pm 0.3}$ | 11.7$_{\pm 0.7}$ | 19.3$_{\pm 2.3}$ | 23.7$_{\pm 0.7}$ | 9.7$_{\pm 0.3}$ | 21.3$_{\pm 2.7}$ | 11.0$_{\pm 0.6}$ | 13.0$_{\pm 1.5}$ | **24.3**$_{\pm 1.8}$ |
| | | 50% | 13.0$_{\pm 0.6}$ | 12.3$_{\pm 0.3}$ | 12.0$_{\pm 0.0}$ | 28.3$_{\pm 2.9}$ | 14.0$_{\pm 0.6}$ | 12.7$_{\pm 0.7}$ | 14.3$_{\pm 0.9}$ | 9.3$_{\pm 0.3}$ | 14.0$_{\pm 2.0}$ | **28.7**$_{\pm 1.4}$ |
| | | 70% | 12.0$_{\pm 0.0}$ | 12.0$_{\pm 0.6}$ | 14.3$_{\pm 0.3}$ | 24.7$_{\pm 0.7}$ | 12.0$_{\pm 1.1}$ | 11.7$_{\pm 0.5}$ | 18.7$_{\pm 2.9}$ | 15.7$_{\pm 0.3}$ | 33.0$_{\pm 1.0}$ | **33.3**$_{\pm 3.8}$ |
| | | 90% | 16.0$_{\pm 0.6}$ | 9.3$_{\pm 0.7}$ | 16.0$_{\pm 0.6}$ | 12.7$_{\pm 0.3}$ | 13.3$_{\pm 0.9}$ | 12.7$_{\pm 0.3}$ | 14.0$_{\pm 0.6}$ | 11.7$_{\pm 0.7}$ | 10.7$_{\pm 0.3}$ | **19.0**$_{\pm 6.1}$ |
| # best-performing tasks | | | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 21 |

## D.2 Further validations of the practical insight using OLMo

We hypothesize that *the effect of exemplar selection diminishes as the model gains more knowledge about the task* and is validated with progressive fine-tuning of Vicuna models in the main paper. In this section, we further supplement the experiments above with empirical evidence using the OLMo models checkpoints [12].

Upon checking the published data sources, named Dolma, of OLMo, it is likely that Instruction Induction (II) data has been included in the training source (via Common Crawl, or The Stack). Therefore, we perform additional experiments across different checkpoints (at 41k, 130k and 410k training steps) of the recent OLMo-7B-0424-hf model, which released checkpoints over more than 400k steps of training. The results are presented below in Tab. 35.

Table 31: Test accuracies counterpart of Tab. 3 over 3 independent trials.

| | EASE | EASE with instructions | improvement |
|---|---|---|---|
| antonyms | **84.7**$_{\pm 0.3}$ | 82.0$_{\pm 0.6}$ | -2.7 ↓ |
| auto_categorization | 34.0$_{\pm 2.1}$ | **48.0**$_{\pm 7.2}$ | 14.0 ↑ |
| diff | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| larger_animal | **88.0**$_{\pm 0.0}$ | 59.3$_{\pm 0.9}$ | -28.7 ↓ |
| negation | **89.0**$_{\pm 0.6}$ | 88.0$_{\pm 0.6}$ | -1.0 ↓ |
| object_counting | 52.0$_{\pm 2.1}$ | **57.3**$_{\pm 3.5}$ | 5.3 ↑ |
| orthography_starts_with | 69.3$_{\pm 0.7}$ | **72.7**$_{\pm 1.2}$ | 3.3 ↑ |
| rhymes | **97.7**$_{\pm 1.3}$ | 68.0$_{\pm 8.5}$ | -29.7 ↓ |
| second_word_letter | 48.3$_{\pm 1.4}$ | **100.0**$_{\pm 0.0}$ | 51.7 ↑ |
| sentence_similarity | **32.7**$_{\pm 1.2}$ | **32.7**$_{\pm 2.7}$ | 0.0 ○ |
| sentiment | 89.0$_{\pm 0.0}$ | **93.7**$_{\pm 0.3}$ | 4.7 ↑ |
| sum | **100.0**$_{\pm 0.0}$ | **100.0**$_{\pm 0.0}$ | 0.0 ○ |
| synonyms | 13.7$_{\pm 0.3}$ | **14.0**$_{\pm 1.5}$ | 0.3 ↑ |
| taxonomy_animal | **78.7**$_{\pm 1.9}$ | 78.3$_{\pm 6.8}$ | -0.3 ↓ |
| translation_en-de | 83.3$_{\pm 0.3}$ | **84.7**$_{\pm 0.3}$ | 1.3 ↑ |
| translation_en-es | 84.7$_{\pm 0.9}$ | **89.3**$_{\pm 0.3}$ | 4.7 ↑ |
| translation_en-fr | 85.0$_{\pm 1.0}$ | **88.0**$_{\pm 0.6}$ | 3.0 ↑ |
| word_sorting | 69.7$_{\pm 0.3}$ | **73.3**$_{\pm 0.7}$ | 3.7 ↑ |
| word_unscrambling | **62.0**$_{\pm 2.1}$ | **62.0**$_{\pm 1.5}$ | 0.0 ○ |
| LR (10% noise) | **42.3**$_{\pm 0.9}$ | 24.0$_{\pm 14.1}$ | -18.3 ↓ |
| LP-variant (10% noise) | 48.0$_{\pm 2.5}$ | **55.0**$_{\pm 1.0}$ | 7.0 ↑ |
| AG News Remap (10% noise) | 30.3$_{\pm 4.4}$ | **51.3**$_{\pm 1.2}$ | 21.0 ↑ |
| SST5 Reverse (10% noise) | 31.3$_{\pm 2.6}$ | **34.0**$_{\pm 1.0}$ | 2.7 ↑ |

Table 32: Test accuracies counterpart of Tab. 4 over 3 independent trials.

**AG News Remap (10% noise)**

| Size $n$ | EASE | EASE with retrieval |
|---|---|---|
| 1000 | 35.3$_{\pm 5.8}$ | **40.7**$_{\pm 0.9}$ |
| 10000 | **39.3**$_{\pm 3.4}$ | 31.3$_{\pm 5.5}$ |
| 50000 | 32.0$_{\pm 1.0}$ | **36.0**$_{\pm 1.0}$ |
| 100000 | 29.0$_{\pm 2.5}$ | **37.3**$_{\pm 2.4}$ |

**SST5 Reverse (10% noise)**

| Size $n$ | EASE | EASE with retrieval |
|---|---|---|
| 1000 | 33.0$_{\pm 2.9}$ | **35.3**$_{\pm 2.6}$ |
| 3000 | **39.0**$_{\pm 1.1}$ | 31.7$_{\pm 2.6}$ |
| 5000 | 36.7$_{\pm 0.9}$ | **37.0**$_{\pm 0.6}$ |
| 7000 | 23.0$_{\pm 2.1}$ | **37.3**$_{\pm 0.9}$ |

The conclusions are consistent with Fig. 1 of the main paper.

- When the training just started (i.e., at 41k steps), the model might not be capable enough to carry out effective in-context learning.
- As the training progresses (i.e., at 130k steps), we observe the best exemplar selection effectiveness. At this point, the model is capable of learning underlying relationships among the in-context exemplars, and yet to be well-trained on the specific task.
- As the model converges (i.e., at 410k steps), the gain from exemplar selection using our EASE diminishes as the model becomes well-trained on the dataset of the respective tasks.

We also tried the rule-based tasks and remapped label tasks on OLMo-7B-0424-hf. However, the in-context learning performances are always at 0% for these more difficult tasks, so comparisons are not meaningful. We also look forward to similar efforts as OLMo in the community to open-source larger and more capable models with checkpoints in the future.

### D.3 Potential limitations of the extension

In Sec. 4.5, we described an extension of EASE using retrieval-based methods to tackle large exemplar sets. Despite the practical effectiveness and efficiency, we elaborate on some limitations here. (1) The filtering through retrieval-based methods completely eliminates the consideration of a large subset of exemplars in the later optimization stage. This may result in important exemplars being left out and


https://doi.org/10.52202/079017-3899

Table 33: The average performance of random exemplar in-context prompting in "out-of-distribution" tasks. It only achieves 17.6% accuracy on average, indicating that the model has little knowledge about the task. The performance gain from EASE is relatively large.

| Task | Noise | Random | EASE | Gap |
|---|---|---|---|---|
| LR | 0% | 34.4±0.0 | 81.7±3.6 | 47.3 |
| | 10% | 28.0±0.1 | 73.3±3.6 | 45.3 |
| | 30% | 18.2±0.1 | 78.3±1.4 | 60.1 |
| | 50% | 10.8±0.1 | 71.7±2.7 | 60.9 |
| | 70% | 4.2±0.1 | 66.7±3.6 | 62.5 |
| | 90% | 0.5±0.0 | 53.3±2.7 | 52.8 |
| LP-variant | 0% | 46.3±0.1 | 75.0±0.0 | 28.7 |
| | 10% | 44.1±0.2 | 75.0±2.4 | 30.9 |
| | 30% | 36.0±0.2 | 73.3±1.4 | 37.3 |
| | 50% | 27.5±0.1 | 76.7±2.7 | 49.2 |
| | 70% | 16.8±0.1 | 75.0±0.0 | 58.2 |
| | 90% | 3.6±0.1 | 63.3±1.4 | 59.7 |
| AG News Remap | 0% | 9.2±0.1 | 53.3±3.6 | 44.1 |
| | 10% | 8.6±0.1 | 56.7±2.7 | 48.1 |
| | 30% | 8.1±0.1 | 51.7±1.4 | 43.6 |
| | 50% | 7.5±0.0 | 56.7±1.4 | 49.2 |
| | 70% | 7.0±0.1 | 51.7±1.4 | 44.7 |
| | 90% | 6.8±0.1 | 55.0±2.4 | 48.2 |
| SST5 Reverse | 0% | 18.8±0.0 | 50.0±0.0 | 31.2 |
| | 10% | 18.2±0.1 | 50.0±0.0 | 31.8 |
| | 30% | 17.5±0.1 | 41.7±3.6 | 24.2 |
| | 50% | 16.9±0.1 | 43.3±1.4 | 26.4 |
| | 70% | 16.8±0.1 | 45.0±2.4 | 28.2 |
| | 90% | 17.3±0.0 | 31.7±1.4 | 14.4 |
| Mean | | 17.6 | 60.4 | 42.8 |

Table 34: The average performance of random exemplar in-context prompting in Instruction Induction (II) tasks. It achieves 64.7% accuracy on average, indicating that the model has much knowledge about the task. The performance gain from EASE is relatively small.

| Task | Random | EASE | Gap |
|---|---|---|---|
| active_to_passive | 100.0±0.0 | 100.0±0.0 | 0.0 |
| antonyms | 79.3±0.0 | 90.0±0.0 | 10.7 |
| auto_categorization | 5.4±0.1 | 30.0±0.0 | 24.6 |
| diff | 6.5±0.0 | 100.0±0.0 | 93.5 |
| first_word_letter | 100.0±0.0 | 100.0±0.0 | 0.0 |
| larger_animal | 83.3±0.1 | 100.0±0.0 | 16.7 |
| letters_list | 100.0±0.0 | 100.0±0.0 | 0.0 |
| negation | 94.7±0.0 | 95.0±0.0 | 0.3 |
| num_to_verbal | 100.0±0.0 | 100.0±0.0 | 0.0 |
| object_counting | 52.9±0.1 | 73.3±1.4 | 20.4 |
| orthography_starts_with | 50.7±0.1 | 78.3±1.4 | 27.6 |
| rhymes | 56.7±0.2 | 100.0±0.0 | 43.3 |
| second_word_letter | 23.8±0.1 | 50.0±0.0 | 26.2 |
| sentence_similarity | 20.5±0.2 | 56.7±1.4 | 36.2 |
| sentiment | 89.8±0.2 | 100.0±0.0 | 10.2 |
| singular_to_plural | 100.0±0.0 | 100.0±0.0 | 0.0 |
| sum | 12.5±0.0 | 100.0±0.0 | 87.5 |
| synonyms | 21.3±0.1 | 30.0±0.0 | 8.7 |
| taxonomy_animal | 51.5±0.2 | 88.3±2.7 | 36.8 |
| translation_en-de | 84.4±0.0 | 90.0±0.0 | 5.6 |
| translation_en-es | 94.1±0.1 | 100.0±0.0 | 5.9 |
| translation_en-fr | 78.7±0.0 | 88.3±1.4 | 9.6 |
| word_sorting | 84.5±0.1 | 91.7±1.4 | 7.2 |
| word_unscrambling | 61.5±0.1 | 78.3±2.7 | 16.8 |
| Mean | 64.7 | 85.0 | 20.3 |

Table 35: Average performance gap between Best-of-N and EASE on different training-step checkpoints of OLMo.

| | OLMo_41k | | | OLMo_130k | | | OLMo_410k | | |
|---|---|---|---|---|---|---|---|---|---|
| | Best-of-N | EASE | Gap | Best-of-N | EASE | Gap | Best-of-N | EASE | Gap |
| object_counting | 20.0±2.9 | 31.7±1.7 | 11.7 | 25.0±2.9 | 35.0±2.9 | 10.0 | 45.0±2.9 | 46.7±1.7 | 1.7 |
| sentence_similarity | 25.0±0.0 | 25.0±0.0 | 0.0 | 30.0±2.9 | 35.0±2.9 | 5.0 | 41.7±1.7 | 45.0±0.0 | 3.3 |
| orthography_starts_with | 21.7±1.7 | 26.7±1.7 | 5.0 | 21.7±1.7 | 26.7±1.7 | 5.0 | 26.7±1.7 | 31.7±1.7 | 5.0 |
| translation_en-fr | 21.7±1.7 | 21.7±1.7 | 0.0 | 38.3±1.7 | 43.3±1.7 | 5.0 | 35.0±0.0 | 40.0±0.0 | 5.0 |

never explored again in our automatic optimization. (2) The cosine similarity retrieval places a strong bias on preferring exemplars that are similar (in the embedding space) to the validation set, which may not yield the best performance in practice. This bias is dependent on the retrieval model and the retrieval metric used which therefore need to be carefully selected. Nevertheless, these limitations come with the simplification of the search space for practical efficiency reasons.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contribution and scope, with bulleted points of contributions nearing the end of the introduction (Sec. 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work have been discussed in the conclusion and limitations (Sec. 6). The computational efficiency of the proposed algorithm is discussed in Sec. 3.2 and App. C.4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?


https://doi.org/10.52202/079017-3899

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code is included in the supplementary materials. In the paper, the main experiment section (Sec. 4) and the section about implementation details (App. B) provide all information needed to reproduce the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code is available at `https://github.com/ZhaoxuanWu/EASE-Prompt-Optimization`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All details necessary to understand the results are provided in the main experiment section (Sec. 4) and the section about implementation details (App. B).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars (that indicates the standard error of the mean) are reported for all results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

 https://doi.org/10.52202/079017-3899

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computer resources utilized to produce the results in the paper are stated in App. B. The amount of compute is compared in App. C.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader societal impacts are discussed in App. A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No model will be released for this work and all datasets involve no risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The sources of the datasets are cited in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

 https://doi.org/10.52202/079017-3899

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new datasets are well documented in Sec. 4 and App. B. The code is well documented as well.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.


https://doi.org/10.52202/079017-3899