Compositional Exemplars for In-context Learning

Jiacheng Ye [1 2] **Zhiyong Wu** [2] **Jiangtao Feng** [2] **Tao Yu** [1] **Lingpeng Kong** [1]

Abstract

Large pretrained language models (LMs) have shown impressive In-Context Learning (ICL) ability, where the model learns to do an unseen task via a prompt consisting of input-output examples as the demonstration, without any parameter updates. The performance of ICL is highly dominated by the quality of the selected in-context examples. However, previous selection methods are mostly based on simple heuristics, leading to sub-optimal performance. In this work, we formulate in-context example selection as a subset selection problem. We propose CEIL (**C**ompositional **E**xemplars for **I**n-context **L**earning), which is instantiated by Determinantal Point Processes (DPPs) to model the interaction between the given input and in-context examples, and optimized through a carefully-designed contrastive learning objective to obtain preference from LMs. We validate CEIL on 12 classification and generation datasets from 7 distinct NLP tasks, including sentiment analysis, paraphrase detection, natural language inference, commonsense reasoning, open-domain question answering, code generation, and semantic parsing. Extensive experiments demonstrate not only the state-of-the-art performance but also the transferability and compositionality of CEIL, shedding new light on in-context learning. Our code is released at https://github.com/HKUNLP/icl-ceil.

1. Introduction

An important goal of artificial intelligence is to develop models that can generalize to unseen tasks. NLP community made a major step towards this goal by discovering the in-context learning (ICL) capability of large pre-trained language models (LMs; Brown et al. 2020). Given a limited number of demonstration examples, in-context learning imitates the human ability to leverage prior knowledge to achieve the best generalization performance.

However, such ability comes along with the robustness issue: ICL is particularly sensitive to the selection of in-context examples, and different arrangements can result in a performance deviation from close to random to near state-of-the-art (Rubin et al., 2022; Liu et al., 2022; Wu et al., 2022). There have been a number of research attempts over the past two years to select better in-context examples. In particular, one prominent approach is to compare the input with each individual example based on learning-free heuristics (Liu et al., 2022) or learning-based metrics (Rubin et al., 2022). Despite the improved performance, these methods do not take into account the inter-relationship between in-context examples. For instance, the ignorance of redundancy of in-context examples can result in almost identical examples, providing no additional supervision. Searching for a compact set of in-context examples becomes even more urgent as there is a hard limit for the prompt length due to the backbone transformer architecture of LMs.

In this paper, we propose a general approach, named CEIL (**C**ompositional **E**xemplars for **I**n-context **L**earning). Instead of selecting each in-context example independently, CEIL models the joint probability of the entire in-context example set, and thus captures the inter-relationship between in-context examples. To model the joint probability of a set given a specific input, we propose a novel model based on the conditional determinantal point process (DPP; Kulesza et al. 2012) that learns to select the most diverse yet helpful in-context example set (§3.1). To take into account the quality of a selected subset, a scoring function from a language model is incorporated into the conditional DPP to form a contrastive loss (§3.2). That way, our algorithm maintains the polynomial time maximum a posteriori (MAP) inference of DPP (Chen et al., 2018) so that the optimal in-context example subset can be found effectively in the inference stage (§3.3).

We validate our method by conducting extensive experiments on 12 classification and generation datasets from 7 distinct tasks, including sentiment analysis, paraphrase detection, natural language inference, commonsense rea-

[1]Department of Computer Science, The University of Hong Kong [2]Shark-NLP, Shanghai Artificial Intelligence Laboratory. Correspondence to: Jiacheng Ye, Zhiyong Wu <carsonye@connect.hku.hk, whucs2013wzy@gmail.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

soning, open-domain question answering, code generation, and semantic parsing. The experiments demonstrate that: 1) CEIL substantially surpasses both conventional learning-free and learning-based selection approaches, achieving state-of-the-art in-context learning performance (§4.4); 2) CEIL shows transferability across LMs and datasets, enabling a learning-free efficient application (§4.6); 3) CEIL inherently learns to compose different examples, shedding new lights on in-context learning for compositional tasks (§4.5); 4) CEIL is especially effective when the number of in-context examples is in a small scale (§4.7).

2. Preliminary

2.1. In-context Learning

In-context learning (ICL) refers to one of the core emergent abilities (Wei et al., 2022) that infers new tasks from context (Brown et al., 2020). We use the terms 'in-weights learning' and 'in-context learning' from prior work on sequence models (Brown et al., 2020) to distinguish between gradient-based learning with parameter updates and gradient-free learning from context, respectively.

Formally, each training instance is first linearized into an input text $\mathbf{x} = (x_1 \dots x_{|\mathbf{x}|})$ and an output text $\mathbf{y} = (y_1 \dots y_{|\mathbf{y}|})$, where for all tokens $x_1 \dots x_{|\mathbf{x}|}, y_1 \dots y_{|\mathbf{y}|} \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary set of the LM. Given a new test input text $\mathbf{x}_{test}$, in-context learning defines the generation of output $\mathbf{y}$ as

$$\mathbf{y}_{test} \sim \mathcal{P}_{LM}(\mathbf{y}_{test} \mid \underbrace{\mathbf{x}_1, \mathbf{y}_1, \dots, \mathbf{x}_K, \mathbf{y}_K}_{context}, \mathbf{x}_{test}),$$

where $\sim$ refers to decoding strategies (e.g., greedy decoding and nuclear sampling (Holtzman et al., 2019)), and each in-context example $e_i = (\mathbf{x}_i, \mathbf{y}_i)$ is sampled from a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. The generation procedure is especially attractive as it eliminates the need for updating the parameters of the language model when encountering a new task, which is often expensive and impractical.

Notably, the performance of ICL on downstream tasks can vary from almost random to comparable with state-of-the-art systems, depending on the quality of the retrieved in-context examples (Rubin et al., 2022; Liu et al., 2022; Wu et al., 2022). Rather than randomly selecting in-context examples for each test input, previous work model the process with a retriever $\mathcal{P}(e_i \mid \mathbf{x}_{test})$, which is either off-the-shelf (Liu et al., 2022; Wu et al., 2022) or fine-tuned (Rubin et al., 2022).

2.2. Determinantal Point Processes

Determinantal point processes (DPPs) are elegant probabilistic models with the ability to express negative interactions. (Kulesza et al., 2012). Formally, a DPP $\mathcal{P}$ is a probability measure for 2^N item sets, where each set consists of items sampled without replacement from a discrete item set $Z = \{1, 2, \dots, N\}$. Given the feature vector $\mathbf{a}$ for each item, DPP calculates an $N \times N$ positive semi-definite (PSD) kernel matrix $\mathbf{L}$, where $\mathbf{L}_{ij} = \mathrm{k}(\mathbf{a}_i, \mathbf{a}_j)$ and $\mathrm{k}(\cdot, \cdot)$ is a kernel function. Then the probability over a subset of items indexed by $S \subseteq Z$ can be defined as

$$\mathcal{P}(S) = \frac{\det(\mathbf{L}_S)}{\det(\mathbf{L} + \mathbf{I})}, \tag{1}$$

where $\mathbf{L}_S \equiv [\mathbf{L}_{ij}]_{i,j \in S}$ denotes the restriction of $\mathbf{L}$ to the entries indexed by elements of S, $\det(\cdot)$ denotes the determinant of a matrix, and $\mathbf{I}$ is an identity matrix. Note according to the the kernel trick (Schölkopf et al., 2002), $\mathrm{k}(\mathbf{a}_i, \mathbf{a}_j)$ can be written as $\phi(\mathbf{a}_i)^T \phi(\mathbf{a}_j)$, where $\phi(\cdot)$ is a reproducing kernel feature map. Therefore, determinants can be geometrically interpreted as the volume of the parallelepiped formed by the vectors $\{\phi(\mathbf{a}_i) \mid i \in S\}$. As the magnitude of an item's feature vector increases, so do the probabilities of sets containing that item. Meanwhile, as the similarity between two items increases, the probabilities of sets containing both of them decrease.

Under the distribution $\mathcal{P}$, although the number of possible realizations of S is exponential in N, many types of inference tasks including marginalization, conditioning, sampling and MAP inference can be performed in polynomial time (Kulesza et al., 2012; Gillenwater et al., 2012; Han et al., 2017; Chen et al., 2018, *inter alia*).

3. Model

In this section, we introduce an efficient framework, CEIL, to learn the **C**omposition of **E**xemplars for **I**n-context **L**earning, as shown in Figure 1. Instead of independently retrieving each in-context example, CEIL models the full in-context example sets by learning the joint probability $\mathcal{P}(S \mid \mathbf{x}_{test})$, and thus captures the inter-relationship between in-context examples. The joint probability is modeled with a learnable conditional DPP (§3.1) and trained with contrastive learning (§3.2). In the inference stage, the best in-context example subset is selected via efficient MAP inference (§3.3).

3.1. Modeling

For in-context learning, both relevance (i.e., choosing in-context examples similar to the test input) and diversity (i.e., the similarity between examples) are essential, while the vanilla DPPs ignore the relevance term. To infuse both relevance and diversity into the selection procedure, we define a new kernel

$$\tilde{\mathrm{k}}(\mathbf{a}_i, \mathbf{a}_j \mid \mathbf{x}) = \mathrm{g}(\mathbf{a}_i, \mathbf{x})\, \mathrm{k}(\mathbf{a}_i, \mathbf{a}_j)\, \mathrm{g}(\mathbf{a}_j, \mathbf{x}), \tag{2}$$

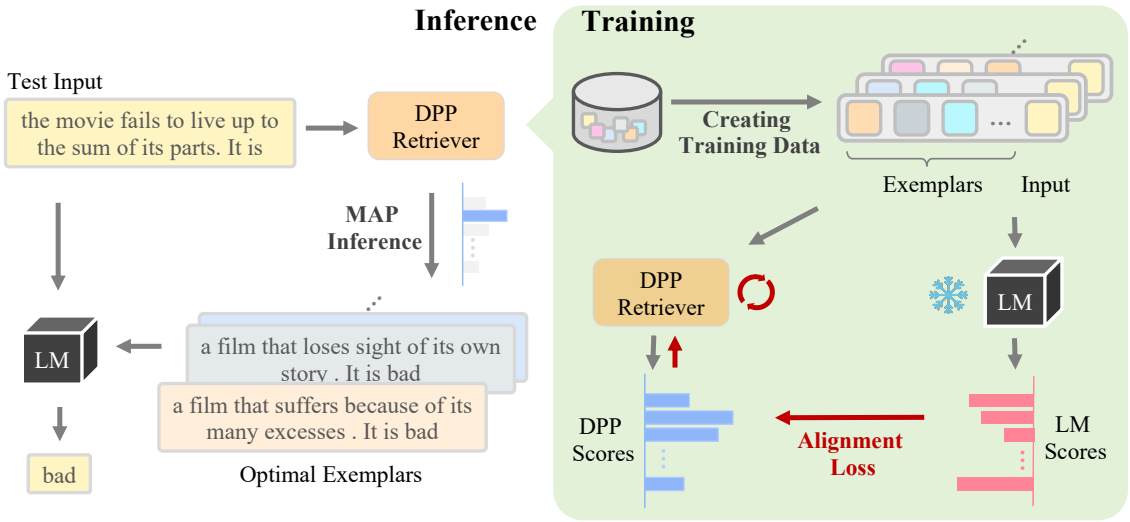

Figure 1. CEIL at training and inference. Instead of independently retrieving each exemplar (or in-context example), CEIL models the entire set of exemplars by learning their joint probability with a conditional DPP (§3.1), which is further trained to align with the LM score through a contrastive loss (§3.2). For a given test input during inference, the optimal exemplar set is obtained by the learned DPP retriever through MAP inference (§3.3). The black-box LM is frozen during the whole procedure.

which is conditioned on the test input $\mathbf{x}$. The new DPP corresponds to a conditional kernel matrix considering both diversity and relevance: $\tilde{\mathbf{L}} = \mathrm{Diag}(\mathbf{r}) \cdot \mathbf{L} \cdot \mathrm{Diag}(\mathbf{r})$, where $\mathbf{r}_i = \mathrm{g}(\mathbf{a}_i, \mathbf{x})$ is the relevance score for item i. Based on Eq. (1) and Eq. (2), we can derive the unnormalized log-probability for subset S as

$$\log\det\left(\tilde{\mathbf{L}}_S\right) = \sum_{i \in S} \log\left(\mathbf{r}_i^2\right) + \log\det\left(\mathbf{L}_S\right),$$

which clearly shows how the DPP model incorporates the relevance (i.e., $\mathbf{r}_i$) and diversity (i.e., $\det(\mathbf{L}_S)$) of the in-context examples.

Intuitively, different tasks may prefer a different trade-off between diversity and relevance, e.g., a more complex input may require a more complicated composition of in-context examples. At the same time, the original DPP model does not offer such a mechanism. To balance the magnitude of diversity and relevance for different tasks, we further incorporate a trade-off parameter λ as follows:

$$\log\det\left(\mathbf{L}'_S\right) = \frac{1}{\lambda} \sum_{i \in S} \mathbf{r}_i + \log\det\left(\mathbf{L}_S\right).$$

This exactly corresponds to a DPP with kernel $\mathbf{L}' = \mathrm{Diag}\left(\exp\left(\frac{\mathbf{r}}{2\lambda}\right)\right) \cdot \mathbf{L} \cdot \mathrm{Diag}\left(\exp\left(\frac{\mathbf{r}}{2\lambda}\right)\right)$.

In practice, the retriever model consists of two embedders to encode input text and in-context examples to their representations $\mathbf{x}$ and $\mathbf{a}$. We set both of the two embedders as highly expressive learnable neural networks (e.g., BERT (Devlin et al., 2019)) such that the resulting DPP score (Eq. (1)) can be an effective ranking metric for subset retrieval. On the high-dimensional embedding space, linear kernel (i.e., dot product) is then applied as similarity function g and k. The learning of the embedder networks essentially becomes a *metric learning* problem (Kulis et al., 2013), which we will introduce in the subsequent section.

3.2. Training

Since there is no ground-truth subset of in-context examples for each training instance, we cannot apply the conventional likelihood-maximization method to learn the parameters. In this section, we introduce a contrastive learning framework, with the main idea of rectifying the embedding of each in-context example and training instance such that a 'better' subset has a higher probability to be retrieved than a 'worse' subset for the training instance.

Training Data. Our goal in construction training data is to obtain a dataset $\mathcal{D}_{train} = \{(e_i, \{S_{ij}, s_{ij}\})_{j=1}^{M}\}_{i=1}^{N}$ consists of N instances. Each instance contains one input instance e_i from the training set $\mathcal{D}$, M in-context example subsets where each example in subset S_{ij} is also retrieved from $\mathcal{D}$[1], and score s_{ij} to indicate the quality of each subset.

Modeling on the full space of S is exponential in N and thus prohibitive. To this end, we employ a two-stage framework which is commonly used in retrieval (Liu et al., 2009). We first precompute a set of relevant examples of size

[1]We omit the retrieved example that is exactly same as input instance e_i to prevent copying answer.

n ($n << N$) with a retriever. Then, we perform non-replacement random sampling to obtain M distinct subsets, with no repeating examples in each subset to prevent zero determinant when calculating $\det(S)$.

Once we retrieve the set of in-context example subsets $\{S_{ij}\}_{j=1}^{M}$ for each input instance $e_i = (\mathbf{x}_i, \mathbf{y}_i)$, we use the inference LM themselves as the scoring function. To measure the quality of each subset, the score is defined as the probability to predict the answer under the LM, which is formally represented as

$$s_{ij} = \mathcal{P}_{LM}\left(\mathbf{y}_i \mid S_{ij}, \mathbf{x}_i\right).$$

This indicates how helpful this subset is for decoding the target answer.

Contrastive Loss. The InfoNCE loss (Oord et al., 2018) has been found effective to learn which single item is superior to others in various representation learning scenarios (Karpukhin et al., 2020; He et al., 2020; Rubin et al., 2022). However, it has the same treatment for all negative samples and the predicted scores s_{ij} are not fully utilized. To mitigate this problem, we propose to employ a fine-grained pair-wise margin loss to determine which subset is preferable, and the loss for each training instance is defined as

$$\mathcal{L}_i = \sum_{(S^+, S^-) \in \mathcal{C}_i} \max\left\{0, \frac{\log \mathcal{P}(S^-) - \log \mathcal{P}(S^+)}{c_i} + \xi\right\}$$
$$c_i = \max_{S \in \mathcal{C}_i} \log \mathcal{P}(S) - \min_{S \in \mathcal{C}_i} \log \mathcal{P}(S),$$

where $\mathcal{C}_i = \{S_{ij}\}_{j=1}^{M}$ contains all the sampled subsets for instance i, ξ is set to $\gamma * (\mathrm{rank}(S^-) - \mathrm{rank}(S^+))$ following (Zhong et al., 2020; An et al., 2022) to reflect the quality difference in these pairs, γ is a hyper-parameter controlling the strength which we set $\gamma = 1/|\mathcal{C}_i|$ such that $\xi \in [0, 1]$, and c_i is used to align the scale with ξ. Note the normalization term $\det(\mathbf{L}+\mathbf{I})$ in Eq. (1) requires calculation with complexity $O(N^3)$ on full items with size N, while the use of pair-wise ranking loss naturally eliminates the calculation of this term (i.e., $\log \mathcal{P}(S^-) - \log \mathcal{P}(S^+) = \log\det\left(\mathbf{L}_{S^-}\right) - \log\det\left(\mathbf{L}_{S^+}\right)$), and thus cuts down the calculation cost.

3.3. Inference

In the inference stage, rather than searching for the most relevant top-k in-context examples as in previous work (Rubin et al., 2022; Liu et al., 2022), we perform maximum a posteriori (MAP) inference with the learned DPP module, considering both diversity and relevance. The MAP inference of a DPP is defined as

$$S_{map} = \arg\max_{S \subseteq Z} \det\left(\mathbf{L}'_S\right),$$

which is NP-hard (Ko et al., 1995). Similar as in constructing training data, we narrow down the candidate space with KNN retriever from N to n. Then we follow Chen et al. (2018) to use an exact implementation of the greedy algorithm with $O(nK^2)$ complexity, where $K = |S_{map}|$ is the number of in-context examples. In each iteration, the example j is greedily selected based on the incremental gain to the log-probability

$$j = \arg\max_{i \in Z \setminus S_{map}} \log\det\left(\mathbf{L}'_{S_{map} \cup \{i\}}\right) - \log\det\left(\mathbf{L}'_{S_{map}}\right).$$

and added to S_{map}. With Cholesky decomposition, the complexity can be reduced from $O(nK^3)$ down to $O(nK)$ in each iteration by updating the Cholesky factor incrementally. Note that compared with vanilla KNN retrieval which directly retrieves K examples from N, the additional inference latency caused by MAP inference is negligible since both n and K here are relatively small numbers (e.g., $n = 100$, $K = 16$).

4. Experiments

We conduct extensive experiments over 12 diverse datasets, spanning 7 distinct tasks, and show a better approach to in-context learning than previously considered.

4.1. Datasets and Evaluation

All the datasets and tasks are listed in Table 1. These datasets involve different task formulations, thereby allowing for extensive evaluations of CEIL in varying scenarios. Prompts and examples of each dataset are shown in Appendix A.1.

We compare the predicted answers with the ground truth and report Accuracy (Acc.) for all the classification tasks. For generation tasks, we report Exact Match (EM) for WebQs, GeoQuery, NL2Bash, MTOP, and SMCalFlow, LF-EM (Hasson & Berant, 2021) for Break following (Rubin et al., 2022), which is an improvement to EM to measure semantically equivalence. Final results are reported on the validation set as the test set is private for some datasets.

4.2. Baselines

Our model CEIL is essentially a learning-based retriever for in-context example selection. We consider both learning-free and other learning-based retrievers as baselines:

- RANDOM: The retriever that randomly selects in-context examples from the training set without repetition.
- TOPK-BM25: The classical sparse retrieval method BM25 (Robertson & Zaragoza, 2009), which is an extension of TF-IDF. Top-K-scored examples are selected as in-context examples.

Table 1. All the datasets and tasks used in the experiments. We show the number of training instances after deduplicating. #ICE refers to the average number of in-context examples for instances in the validation set when using GPT-Neo as LLM.

Type	Dataset	Task	#Train	#Validation	#ICE
Classification	SST-5 (Socher et al., 2013)	Sentiment Analysis	8,534	1,101	40
	MRPC (Dolan et al., 2004)	Paraphrase Detection	3,668	408	27
	MNLI (Williams et al., 2018)	Natural Language Inference	392,568	19,647	40
	QNLI (Wang et al., 2018)	Natural Language Inference	104,707	5,463	27
	CMSQA (Talmor et al., 2019)	Commonsense Reasoning	9,740	1,221	50
	HellaSwag (Zellers et al., 2019)	Commonsense Reasoning	52,611	20,006	50
Generation	WebQs (Berant et al., 2013)	Open-Domain QA	3,778	2,032	50
	GeoQuery (Zelle & Mooney, 1996)	Code Generation	404	280	50
	NL2Bash (Lin et al., 2018)	Code Generation	7,441	609	43
	Break (Wolfson et al., 2020)	Semantic Parsing	44,184	7,760	28
	MTOP (Li et al., 2021)	Semantic Parsing	15,564	2,235	41
	SMCalFlow (Andreas et al., 2020)	Semantic Parsing	102,491	14,751	22

- TOPK-BERT: The dense retriever based on BERT embeddings (Devlin et al., 2019), we adopt `bert-base-uncased`[2] which is publically available in Huggingface Transformers (Wolf et al., 2020).
- DPP-BERT: The DPP retriever directly uses the original BERT embedding as above without fine-tuning, and adopts MAP inference for subset retrieval (Chen et al., 2018).
- TOPK-CONTRIEVER and TOPK-SIMCSE: Two better sentence embedding models trained with contrastive learning (Izacard et al., 2021; Gao et al., 2021b).
- EPR: The learning-based dense retriever trained to retrieve a better singleton in-context example (Rubin et al., 2022), and Top-K most similar examples are selected in the inference stage. We extend it to other tasks beyond semantic parsing in Rubin et al. (2022).

4.3. Implementation Details

We mainly use GPT-Neo (Black et al., 2021) as LLM, which is a 2.7B-parameter LM trained on The Pile (Gao et al., 2021a), an 825 GB text corpus constructed from a wide range of high-quality resources. We also consider GPT2-XL (Radford et al., 2019) (1.5B) and Codex (Chen et al., 2021b) (175B) in §4.6. The number of in-context examples is set to 50, and we truncate it based on the maximum context size for different LMs (e.g., 1,024 for GPT2-XL, 2,048 for GPT-Neo, and 8,001[3] for Codex) on each task. The resulting average number of in-context examples for each task are listed in Table 1.

We sort exemplars based on their similarities to the input text in ascending order, in accordance with common practices (Rubin et al., 2022; Qiu et al., 2022b; Levy et al., 2022). During answer generation, all the classification tasks are reframed into multiple choice following (Brown et al., 2020). We provide the context plus an answer option as input to LM, compare the LM likelihood of each option, and choose the one with the maximum likelihood as the answer. On tasks that involve multi-label classification, each label is given a semantically meaningful name as an option (e.g. "Positive" or "Negative" rather than 0 or 1 for sentiment analysis), and then treat the task like multiple choice. For generation tasks, we use greedy decoding to generate answers.

When constructing data for training the retriever, we limit the number of instances to 44,000 following (Rubin et al., 2022) to reduce the scoring cost, and we sample 50 candidate subsets with 16 examples in each subset for each training instance. We use Adam optimizer (Kingma & Ba, 2015) with batch size 128 and learning rate 1e-5, and run training for 30 epochs on two NVIDIA A100 GPUs. For each task, we search the trade-off factor λ in $\{0.01, 0.05, 0.1\}$. To encode each example into embeddings, we concatenate all the texts in an instance except labels (e.g., premise plus hypothesis in NLI tasks) as input to the BERT-based encoder (i.e., BERT-base with 110M learnable parameters). We initialize the encoder with EPR, which we find significantly helps in training CEIL (§4.7).

4.4. Main Results

We experiment on 12 datasets spanning 7 distinct tasks and the results are shown in Table 2. Overall, we found generation tasks benefit more from a better set of in-context examples than classification tasks. For example, the simple TOPK-BM25 retriever brings an around 12% to

[2]https://huggingface.co/bert-base-uncased
[3]https://platform.openai.com/docs/models/codex

Table 2. Main results on various datasets. We show the absolute performance gain over EPR and **bold** the best results.

Method	SST-5	MRPC	QNLI	MNLI	CMSQA	HellaSwag	WebQs	GeoQ.	NL2Bash	Break	MTOP	SMCal.	Avg.
Learning-free													
RANDOM	31.43	67.65	56.67	37.74	**42.51**	41.16	4.87	33.93	34.35	1.70	7.30	8.90	30.68
TOPK-BM25	36.06	69.36	62.29	40.68	36.12	42.20	16.68	62.86	58.98	26.00	52.70	46.10	45.84
TOPK-CONTRIEVER	37.06	67.89	60.97	45.28	36.12	41.60	17.62	68.93	53.69	26.34	49.84	43.44	45.73
TOPK-SIMCSE	37.06	66.91	61.58	44.85	35.54	41.69	16.83	66.43	54.89	26.58	47.29	42.59	45.19
TOPK-BERT	37.24	69.36	64.65	42.15	35.38	40.28	17.08	66.79	51.30	26.84	52.13	44.63	45.65
DPP-BERT	36.78	69.61	63.83	39.60	37.26	40.69	14.57	70.71	48.99	26.70	53.14	43.26	45.43
Learning-based													
EPR	42.82	75.98	80.76	66.06	36.77	42.61	19.59	68.57	56.82	31.90	64.20	54.30	53.37
CEIL	**47.05**	**80.15**	**85.41**	**71.74**	37.18	**43.20**	**20.92**	**73.21**	**59.91**	**34.18**	**67.43**	**60.73**	**56.76**
Δ Absolute gain	+4.23	+4.17	+4.65	+5.68	+0.41	+0.59	+1.33	+4.64	+3.09	+2.28	+3.23	+6.43	+3.39

Table 3. Results on compositional semantic parsing datasets using GPT-Neo and Codex as inferencers. The retriever used for Codex is the same as that for GPT-Neo, and is trained on the GeoQuery and SMCalFlow datasets. 0-S referring to a non-compositional test set and k-C referring to a compositional test set with additional k-shot compositional examples as demonstrations ($k \in \{0, 8, 16, 32\}$; see Appendix A for details). We show the absolute performance gain over EPR and **bold** the best results.

Model	GeoQuery				SMCalFlow-CS				
	Standard	Template	TMCD	Length	0-S	0-C	8-C	16-C	32-C
Previous Results									
T5 Base + CSL-Aug (Qiu et al., 2022a)	93.30	89.30	74.90	67.80	(Different Dataset Version)				
Cover-LS (Levy et al., 2022)	91.40	81.60	76.30	70.00					
PaLM 540B (Qiu et al., 2022b)	86.80	76.60	63.60	57.90	-	-	4.70	5.00	11.70
PaLM 540B (Oracle) (Qiu et al., 2022b)	92.10	77.93	73.83	63.90	-	-	33.90	36.70	45.60
GPT-Neo 2.7B Inferencer									
TOPK-BERT	66.79	30.75	41.82	31.59	31.94	0.00	0.28	-	-
EPR	68.57	38.95	44.09	32.27	57.78	0.00	0.00	-	-
CEIL	**73.21**	**40.77**	**44.09**	**32.73**	**60.27**	0.00	**0.28**	-	-
Δ Absolute gain	+4.64	+1.82	+0.00	+0.46	+2.49	+0.00	+0.28	-	-
Codex 175B Inferencer									
TOPK-BERT	91.79	87.47	61.36	69.55	80.83	0.00	40.83	46.67	49.72
EPR	91.70	87.93	62.73	73.41	80.83	0.56	35.56	38.61	48.06
CEIL	**93.21**	**89.98**	**63.64**	**74.09**	**81.39**	**1.67**	**42.78**	**48.06**	**55.28**
Δ Absolute gain	+1.51	+2.50	+0.91	+0.68	+0.56	+1.11	+7.22	+9.45	+7.22

45% absolute performance gain compared to the RANDOM retriever. The underlying reason can be that relevant answers rarely appear in the non-relevant exemplars for the generation tasks.

We find CEIL substantially outperforms learning-free baselines and is especially effective on Natural Language Inference (NLI) tasks (e.g., QNLI, MNLI), where more than 20% absolute improvements are obtained. On most of the other classification and generation tasks, CEIL surpasses the learning-free retrievers by around 10%, with an exception on Commonsense Reasoning tasks (i.e., CMSQA and HellaSwag). Interestingly, all the other retrievers (e.g., TOPK-BM25, TOPK-BERT and EPR) perform comparable to the random retriever on this task, indicating the related commonsense knowledge may not exists in the training data.

Compared with the learning-based retriever, CEIL consistently outperforms EPR on all the tasks, suggesting the effectiveness of bringing interaction between in-context examples into the learning procedure. Note CEIL introduces no additional parameters compared with EPR and the learning-free TOPK-BERT, suggesting CEIL is not only effective but also can be efficiently applied in real applications with no deployment cost.

4.5. Compositionality

A natural intuition of the superior performance of CEIL is that it learns to compose exemplars such that the whole subset helps in predicting answers. To systematically investigate the compositional ability of the learned retriever, we experiment on two well-designed semantic parsing datasets obtained from original SMCalFlow and GeoQuery datasets,

where the test examples requires explicit compositional exemplars (e.g., to predict the program of "organize an event with my manager", one has to retrieve exemplars relates to "organize an event" and "my manager"). We evaluate the trained retrievers in §4.4 on various data splits in these two datasets (see Appendix A for details), and the results are shown in Table 3.

The `Template` and `Standard` splits account for the majority of the performance difference between CEIL and EPR, with around 2% and 5% on GeoQuery dataset. Meanwhile, the improvement on all the cross-domain splits (k-C) of SMCalFlow-CS excel the single-domain split (0-S) when comparing CEIL with TOPK-BERT and EPR. These indicate CEIL does, to a certain extent, retrieve compositional exemplars. Overall, CEIL improves performance on all the difficult splits on these two datasets, indicating better organizing the in-context examples helps in predicting compositional and longer target programs.

The previous solutions to generating compositional programs require compositional data augmentation for training LM (Qiu et al., 2022a), or test-time local-structure prediction for selecting diverse exemplars (Levy et al., 2022). CEIL can be seen as an alternative approach that directly retrieves a diverse exemplars subset without tuning inference LM, which is expensive, or test-time question decomposition, which impairs efficiency and may suffer from error propagation. Note though the inference LM in CEIL hasn't seen any compositional data in the context, the retriever has seen as it needs to be trained in the standard dataset. An interesting further work would be training a retriever that directly generalizes to unseen compositional tasks without seeing any compositional data, as we have shown the possibility of transferring across datasets in §4.6.

4.6. Transferability

The compositional characteristics of natural language are general, meaning the retriever may exploit similar knowledge in different tasks or inference LMs. This motivates us to explore whether the retriever trained on one dataset and LM inferencer can be directly transferred to others without further tuning. This is a practical research question as training a retriever for each dataset or LM inferencer can be costly in real applications.

Transfer across LMs We consider transferring the retriever trained on GPT-Neo to a similar-sized model GPT2-XL (Radford et al., 2019) (1.5B) and a much larger model Codex (Chen et al., 2021b) (175B). Note in the transfer setting, CEIL becomes a learning-free method under the target LM, thus we also compare the results with TOPK-BERT. We show the absolute improvement over TOPK-BERT in Figure 2 (Left). Interestingly, the retriever learned

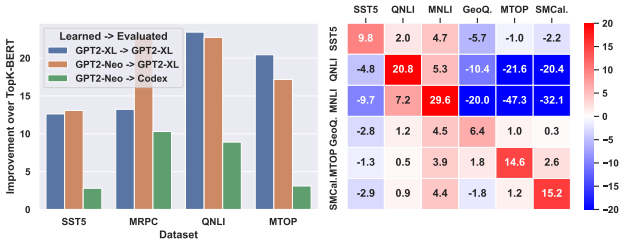

Figure 2. **(Left)** Results of transferring a retriever learned on one LM inferencer to others. **(Right)** Results of transferring a retriever learned on one dataset (row) to others (column). For both figures, we show the absolute improvement over TOPK-BERT.

on GPT2-Neo performs comparably with that on GPT2-XL when evaluating on GPT2-XL for datasets such as SST5, QNLI, and MTOP. We also surprisingly find the transferred retriever outperforms the specially-trained one on the MRPC dataset, indicating it may bring extra knowledge (e.g., compositional characteristic of natural language) beyond learning from the target LM. Note when considering a large LM (e.g., Codex) as the LM inferencer, learning an LM-specific retriever can be costly due to the restricted access. Though TOPK-BERT already performs well on Codex, CEIL still brings improvement.

Transfer across Datasets We further investigate whether a retriever trained on one dataset transfers to others, as shown in Figure 2 (Right). We find almost all the retrievers transfer to NLI tasks such as QNLI and MNLI, and achieve better performance than TOPK-BERT. However, the NLI-trained retrievers hardly transfer to other tasks except for NLI task (e.g., QNLI-trained retriever only benefits MNLI). We conjecture that this is due to the fact that NLI tasks require two text inputs, but other tasks only require one, and that knowledge gained from single-input tasks still has value in double-input tasks. For other single input tasks, we find only the retriever learned on similar tasks (e.g., Code Generation and Semantic Parsing) shows transferability. Developing a retriever works for all tasks is a challenging but valuable research topic, which we leave for future work.

4.7. Analysis

On the Effect of Training Data To investigate the effect of training data, we compare different candidate sampling strategies and the number of candidates. Beyond sampling candidates randomly, we also sample fix-sized candidates based on probability defined by k-DPP (Kulesza & Taskar, 2011). We always include the Top-K candidate, thus we also report $MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$ to measure the quality of the training data based on the ranking of the Top-K candidate among all the candidates. A lower MRR means that there are more candidates that are "better" than the Top-K. As shown

Table 4. Results of various sampling strategies and number of candidates (C) per instance in construction training data. We report both MRR of the Top-K candidate and the performance of the trained retriever.

Method	SST5	MRPC	GeoQuery	MTOP
RAND, C50	0.08/35.97	0.08/80.88	0.08/71.07	0.07/56.60
TOP100+RAND, C10	0.29/46.14	0.29/**81.37**	0.27/67.86	0.25/62.37
TOP100+RAND, C50	0.09/**47.05**	0.09/80.15	0.08/**73.21**	0.09/**67.43**
TOP100+K-DPP, C50	0.09/45.96	0.09/79.41	0.09/71.07	0.09/63.62

Table 5. Comparisons of different initializations and contrastive losses for CEIL.

Method	SST5	MRPC	QNLI	GeoQuery	MTOP
Baselines					
TOPK-BERT	37.24	69.36	64.65	66.79	52.13
EPR	42.82	75.98	80.76	68.57	64.20
Training Strategies					
BERT INIT + INFONCE	31.34	69.12	63.92	68.57	47.43
BERT INIT + PAIR-WISE	35.55	67.89	65.00	67.50	41.30
EPR INIT + INFONCE	**49.14**	**80.64**	**85.54**	69.29	61.92
EPR INIT + PAIR-WISE	47.05	80.15	85.41	**73.21**	**67.43**

in Table 4, the one-stage random retrieval greatly degrades performance on SST5 and MTOP datasets. Surprisingly, the MRR of one-stage random retrieval achieves the lowest, indicating relevance is not the only factor that contributes to the quality of a subset. Two-stage random sampling slightly outperforms k-DPP sampling with similar MRR. Furthermore, we find the number of candidates mostly affects generation tasks, which is considered to be more complex than classification and increasing the number improves the final performance.

On the Effect of Learning Strategies We compare different initializations and contrastive losses in Table 5. Learning which subset is superior based on the raw BERT encoders is challenging, but using EPR as an initializer greatly improves performance. This indicates the knowledge learned from a single in-context example selection contributes to the set-level selection. Regarding the choice of contrastive loss, we find InfoNCE and pair-wise margin loss perform comparably on classification tasks, but the latter significantly surpasses the former on generation tasks, with approximately 4% and 6% on GeoQuery and MTOP, respectively. Note that generation tasks are more difficult than classification as the answers rarely appear in the in-context examples directly. This indicates pair-wise margin loss, which is a more fine-grained contrastive loss than InfoNCE loss, better displays its effectiveness on much harder tasks.

On the Effect of Inference Strategies In this paragraph, we compare two inference algorithm (i.e., TOPK and DPP (short for DPP-MAP)) across learning-free and learning-based methods. Compared with TOPK, we find DPP-MAP brings more improvement when using a learning-based retriever, indicating the importance of aligning the 'similarity' of embedding to the 'usefulness' for inference. Beyond accuracy, we also find the latency of retrieving 50 in-context examples for TOPK and DPP-MAP on SST5 dataset are 30s and 36s (1.2x), respectively. Thus, we recommend choosing TOPK or DPP-MAP for different tasks considering the additional inference cost in real applications. We provide more details on the performance-efficiency trade-off in Appendix

Table 6. Comparison of inference algorithms, i.e., TOPK and DPP (short for DPP-MAP)), on BERT, EPR and CEIL.

Method	SST5	MRPC	MNLI	CMSQA	MTOP	SMCal.
learning-free						
TOPK-BERT	37.24	69.36	42.15	35.38	52.13	44.63
DPP-BERT	36.78	69.61	39.60	37.26	53.14	43.26
learning-based						
TOPK-EPR	42.82	75.98	66.06	36.77	64.20	54.30
DPP-EPR	45.54	80.39	65.09	35.54	64.38	57.64
TOPK-CEIL	45.78	**81.37**	71.25	37.10	66.62	59.95
DPP-CEIL	**47.05**	80.15	**71.74**	**37.18**	**67.43**	**60.73**

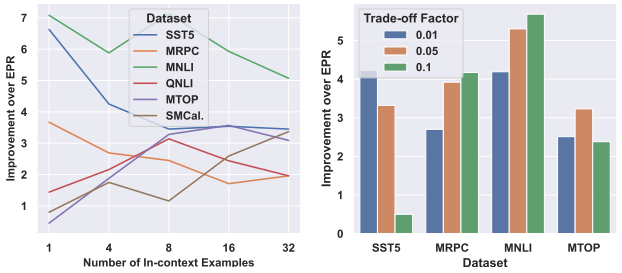

Figure 3. **(Left)** Comparison of different number of in-context examples on various datasets.**(Right)** Comparison of different trade-off factors on various datasets. For both figures, we show the absolute improvement over EPR.

On the Effect of In-context Example Numbers Most of the current large LMs are trained with a limited input length such as 1,024 in GPT2-XL and 2,048 in GPT2-Neo, which restricts the maximum number of in-context examples. Here we evaluate the trained retriever under various number of in-context examples, as shown in Figure 3 (Left). We find a clear increasing trend for most classification tasks when decreasing the numbers, indicating the effectiveness in selecting a compact set of in-context examples. We observe an opposite trend in generation tasks, which we hypothesize is because the difficulty of generation tasks. i.e., the question can only be answered with a sufficient number of in-context examples. Another advantage of a compact set of in-context examples is we can greatly cut down the computations, as the attention module (Vaswani et al., 2017) in most LMs is of quadratic complexity. We

find CEIL mostly outperforms EPR and TOPK-BERT with 32 in-context examples by using merely 4 and 1 example, respectively (see Appendix B.2 for details).

On the Effect of Trade-off Factor We perform an ablation study to see the effect of trade-off factor in Figure 3 (Right). Note a smaller factor put more emphasize on the relevance. We find the best performing factor varies for different datasets. A general observation is that diversity is more important for more difficult tasks, such as NLI and semantic parsing, but relevance is more crucial for the simpler tasks such as sentiment analysis. Given the discrepancy, we find introducing the trade-off factor still consistently outperforms EPR baselines that only considers relevance, verifying the effectiveness of CEIL.

5. Related Work

5.1. In-context Learning

By providing a few input-output examples as demonstrations, in-context learning (ICL) empowers large language models (LMs) to "learn by analogy" and perform complex tasks such as web browsing (Nakano et al., 2021), coding (Chen et al., 2021a), data generation (Ye et al., 2022a; 2023), strategic game (FAIR et al., 2022), and conversations (OpenAI, 2022). The popularity of ICL also raises growing concerns regarding its instability: given different selections, ICL's performance can vary from near state-of-the-art to random (Liu et al., 2022). To mitigate this issue, researchers have made significant efforts on in-context example selection, which can be cataloged into *learning-free* and *learning-based* methods. In the line of learning-free methods, various heuristic criteria are proposed, such as the semantic similarity between testing examples and demonstrations (Liu et al., 2022), entropy (Lu et al., 2022; Wu et al., 2022), diversity (Ye et al., 2022b; Su et al., 2022; Levy et al., 2022; Agrawal et al., 2022). However, learning-free methods generally require human experts to design task-specific heuristics and lead to sub-optimal performance. Researchers thus have started to explore learning-based methods to push the envelope further. Rubin et al. (2022) propose to train a singleton example scorer using contrastive learning with signals from LM inferencer. In comparison, we aim to jointly model the selection of the entire exemplar set, which additionally considers the interaction between in-context examples. Beyond in-context example selection, some works have explored multi-pass ICL, which first generates multiple responses from various subsets of exemplars (Shi et al., 2022; Li et al., 2022) and then aggregate them through techniques similar to self-consistency (Wang et al., 2022). In contrast, multi-pass ICL approaches require multiple test-time inferences, which can result in inefficiency.

5.2. Determinantal Point Processes

Determinantal point processes (DPPs) are efficient probabilistic models that can measure both the diversity and quality of items in a subset, which makes it a natural choice for the diverse subset selection problem (Kulesza et al., 2012). DPPs have been applied for document and video summarization (Kulesza & Taskar, 2011; Gong et al., 2014), recommendation systems (Gillenwater et al., 2012), object detection (Azadi et al., 2017) and multi-label classification (Xie et al., 2017). Most recently, DPPs have been employed in in-context learning specially for compositional tasks (Levy et al., 2022), where the authors first predict all possible target subphrases with a specially-trained model, and then adopt DPPs to sample a diverse subset of in-context examples to cover as many subphrases as possible. However, the diversity objective in DPPs is not aligned with LMs and is generally task-specific. In contrast, we frame DPPs into an end-to-end framework, which not only captures the interaction between in-context examples but also well reflects the preference of LMs on the probability of DPPs.

6. Conclusion

In this paper, we recast in-context example selection into an end-to-end optimization problem. We propose CEIL, which leverages DPP to model the probability of the entire subset of in-context examples, and is learned through a contrastive learning framework. Results on 7 classification and generation tasks with 12 different benchmarks show that CEIL clearly beats previous competitive methods. The learned retriever in CEIL also exhibits surprising transferability across LMs and datasets, and compositionality for compositional tasks, showing an effective and efficient approach to adapt the black-box large LMs to the downstream tasks.

Acknowledgement

We thank the anonymous reviewers whose suggestions helped clarify this work. This work is partially supported by the Shanghai Committee of Science and Technology (Grant No. 21DZ1100100), and the joint research scheme of the National Natural Science Foundation of China (NSFC) and the Research Grants Council (RGC) under grant number N_HKU714/21.

References

Agrawal, S., Zhou, C., Lewis, M., Zettlemoyer, L., and Ghazvininejad, M. In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*, 2022.

An, C., Feng, J., Lv, K., Kong, L., Qiu, X., and Huang, X. Cont: Contrastive neural text generation. *NeurIPS*, 2022.

Andreas, J., Bufe, J., Burkett, D., Chen Jr, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., et al. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020.

Azadi, S., Feng, J., and Darrell, T. Learning detection with diverse proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7149–7157, 2017.

Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1160.

Black, S., Gao, L., Wang, P., Leahy, C., and Biderman, S. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

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

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021a.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021b.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dolan, W. B., Quirk, C., and Brockett, C. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pp. 350–356, 2004.

FAIR, Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science (New York, NY)*, 378(6624):1067–1074, 2022.

Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., and Radev, D. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 351–360, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1033. URL https://aclanthology.org/P18-1033.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv preprint*, abs/2101.00027, 2021a. URL https://arxiv.org/abs/2101.00027.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021b.

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

Gong, B., Chao, W.-L., Grauman, K., and Sha, F. Diverse sequential subset selection for supervised video summarization. *Advances in neural information processing systems*, 27, 2014.

Han, I., Kambadur, P., Park, K., and Shin, J. Faster greedy map inference for determinantal point processes. In

International Conference on Machine Learning, pp. 1384–1393. PMLR, 2017.

Hasson, M. and Berant, J. Question decomposition with dependency graphs. In *3rd Conference on Automated Knowledge Base Construction*, 2021.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Heilbron, F. C., Escorcia, V., Ghanem, B., and Niebles, J. C. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 961–970. IEEE, 2015.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

Ko, C.-W., Lee, J., and Queyranne, M. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

Kulesza, A. and Taskar, B. k-dpps: Fixed-size determinantal point processes. In *ICML*, 2011.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Kulis, B. et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Levy, I., Bogin, B., and Berant, J. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*, 2022.

Li, H., Arora, A., Chen, S., Gupta, A., Gupta, S., and Mehdad, Y. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2950–2962, 2021.

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*, 2022.

Lin, X. V., Wang, C., Zettlemoyer, L., and Ernst, M. D. Nl2bash: A corpus and semantic parser for natural language interface to the linux operating system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation LREC 2018, Miyazaki (Japan), 7-12 May, 2018.*, 2018.

Liu, J., Shen, D., Zhang, Y., Dolan, W. B., Carin, L., and Chen, W. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, 2022.

Liu, T.-Y. et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3): 225–331, 2009.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI, T. Chatgpt: Optimizing language models for dialogue. *OpenAI*, 2022.

Qiu, L., Shaw, P., Pasupat, P., Nowak, P., Linzen, T., Sha, F., and Toutanova, K. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4341–4362, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.323. URL https://aclanthology.org/2022.naacl-main.323.

Qiu, L., Shaw, P., Pasupat, P., Shi, T., Herzig, J., Pitler, E., Sha, F., and Toutanova, K. Evaluating the impact of model scale for compositional generalization in semantic parsing. *arXiv preprint arXiv:2205.12253*, 2022b.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Robertson, S. and Zaragoza, H. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389, 01 2009. doi: 10.1561/1500000019.

Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., and Schiele, B. Movie description. *International Journal of Computer Vision*, 123:94–120, 2017.

Rubin, O., Herzig, J., and Berant, J. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191.

Schölkopf, B., Smola, A. J., Bach, F., et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Shaw, P., Chang, M.-W., Pasupat, P., and Toutanova, K. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 922–938, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.75. URL https://aclanthology.org/2021.acl-long.75.

Shi, F., Fried, D., Ghazvininejad, M., Zettlemoyer, L., and Wang, S. I. Natural language to code translation with execution. *arXiv preprint arXiv:2204.11454*, 2022.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Su, H., Kasai, J., Wu, C. H., Shi, W., Wang, T., Xin, J., Zhang, R., Ostendorf, M., Zettlemoyer, L., Smith, N. A., et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wolfson, T., Geva, M., Gupta, A., Gardner, M., Goldberg, Y., Deutch, D., and Berant, J. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020.

Wu, Z., Wang, Y., Ye, J., and Kong, L. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*, 2022.

Xie, P., Salakhutdinov, R., Mou, L., and Xing, E. P. Deep determinantal point process for large-scale multi-label classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Ye, J., Gao, J., Wu, Z., Feng, J., Yu, T., and Kong, L. ProGen: Progressive zero-shot dataset generation via in-context feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3671–3683, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.269.

Ye, J., Li, C., Kong, L., and Yu, T. Generating data for symbolic language with large language models. 2023.

Ye, X., Iyer, S., Celikyilmaz, A., Stoyanov, V., Durrett, G., and Pasunuru, R. Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*, 2022b.

Yin, P., Fang, H., Neubig, G., Pauls, A., Platanios, E. A., Su, Y., Thomson, S., and Andreas, J. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2810–2823, 2021.

Zelle, J. M. and Mooney, R. J. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pp. 1050–1055, Portland, OR, August 1996. AAAI Press/MIT Press. URL http://www.cs.utexas.edu/users/ai-lab?zelle:aaai96.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6197–6208, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.552. URL https://aclanthology.org/2020.acl-main.552.

A. Experimental Setup

A.1. Datasets

We conduct experiments on 12 classification and generation tasks, and examples in each dataset are shown in Table 7. We illustrate the detail of each dataset as follows.

SST-5 (Socher et al., 2013) is a sentiment classification benchmark containing five fine-grained classes including 'very positive', 'positive' 'neutral', 'negative', and 'very negative'.

MRPC (Dolan et al., 2004) is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent.

MNLI (Williams et al., 2018) is a crowdsourced collection of sentence pairs with textual entailment annotations. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral).

QNLI (Wang et al., 2018) is a question-answering dataset consisting of question-paragraph pairs, and the task is to determine whether the context sentence contains the answer to the question.

CMSQA (Talmor et al., 2019) (short for CommonsenseQA) is a multiple-choice question-answering dataset that requires different types of commonsense knowledge. The task is to predict the correct answer out of five provided candidate answers.

HellaSwag (Zellers et al., 2019) is a large-scale dataset of grounded commonsense reasoning. There are four candidate answers for each question: a video caption from ActivityNet Captions (Heilbron et al., 2015) and the Large Scale Movie Description Challenge (Rohrbach et al., 2017). The three incorrect answers are adversarially generated and human validated to deceive machines. The correct answer is the actual video caption for the subsequent occurrence in the video.

WebQs (Berant et al., 2013) (short for WebQuestions) is question-answer pairs obtained from the web. The questions are selected using Google Suggest API, and the answers are entities in Freebase.

NL2Bash (Lin et al., 2018) is a dataset for the problem of mapping English sentences to Bash commands. The corpus consists of text–command pairs, where each pair

Table 7. Datasets with corresponding prompts and examples used in the experiments.

Dataset	Prompt	Example
SST-5	{input} It is {output}	**Input**: this is a stunning film , a one-of-a-kind tour de force . **Output**: very positive
MRPC	{input1} Can we say "{input2}"? {output}	**Input1**: The company didn 't detail the costs of the replacement and repairs. **Input2**: But company officials expect the costs of the replacement work to run into the millions of dollars . **Output**: No
MNLI	{input1} Can we say "{input2}"? {output}	**Input1**: yeah i know and i did that all through college and it worked too **Input2**: I did that all through college but it never worked **Output**: No
QNLI	{input1} Can we know "{input2}"? {output}	**Input1**: As of that day, the new constitution heralding the Second Republic came into force. **Input2**: What came into force after the new constitution was herald? **Output**: Yes
CMSQA	{input} {output}	**Input**: Sammy wanted to go to where the people were. Where might he go? **Output**: populated areas
HellaSwag	{input} {output}	**Input**: Members of the procession walk down the street holding small horn brass instruments. A drum line **Output**: passes by walking down the street playing their instruments
WebQs	{input} {output}	**Input**: what does jamaican people speak? **Output**: Jamaican Creole English Language
GeoQuery	{input}\t{output}	**Input**: what is the population of montana ? **Output**: answer(A,(population(B,A),const(B,stateid(montana))))
NL2Bash	{input}\t{output}	**Input**: find all executable files in /home directory. **Output**: find /home -type f -perm /a=x
Break	{input}\t{output}	**Input**: How many large metallic items are there? **Output**: 1#) return items 2#) return #1 that are large 3#) return #2 that are metallic 4#) return number of #3
Mtop	{input}\t{output}	**Input**: Resume the timer in 10 seconds **Output**: [IN:RESUME_TIMER [SL:METHOD_TIMER timer] [SL:DATE_TIME in 10 seconds]]
SMCalFlow	{input}\t{output}	**Input**: Can you create me a new meeting on thursday morning? **Output**: (Yield (CreateCommitEventWrapper (CreatePreflightEventWrapper (Event.start_? (DateTimeConstraint (Morning) (NextDOW (Thursday)))))))

consists of a Bash command scraped from the web and an expert-generated natural language description.

GeoQuery (Zelle & Mooney, 1996; Shaw et al., 2021) contains a parallel corpus of 880 English questions about US geography paired with Prolog queries. The compositional dataset of GeoQuery were created by Shaw et al. (2021), focusing on compositional generalization. In addition to the original `Standard` split, it contains three additional splits: (1) the `Template` split, where abstract output templates in training and test data are disjoint (Finegan-Dollak et al., 2018); (2) the `TMCD` split, which makes the distributions of compounds in training and test data as divergent as possible; and (3) the `Length` split, where the test instances are longer than the training ones.

Break (Wolfson et al., 2020) is a dataset that maps complex natural language questions into a language-based meaning representation. The question is decomposed into an ordered list of atomic steps, which is used as the target sequence. We use the low-level Break subset following (Rubin et al., 2022).

MTOP (Li et al., 2021) is a multilingual task-oriented semantic parsing dataset covering 6 languages and 11 domains. The target commands are complex queries featuring nested intent-slot prediction. Similar to past work (Rubin et al., 2022), we use the English subset of MTOP.

SMCalFlow (Andreas et al., 2020; Yin et al., 2021) is a large dialogue dataset, featuring natural conversations about tasks involving calendars, weather, places, and people. The meaning representation is an executable dataflow program featuring API calls, function composition, and complex constraints. The SMCalFlow-CS (Yin et al., 2021) dataset is a subset of SMCalFlow, containing single-turn natural sentences involving two domains (organization structure and event creation), each having its own set of program symbols. The cross-domain (C) test set evaluates examples that incorporate compositional abilities, while the single-domain (S) test set contains examples from a single domain. On few-shot settings (split k-C, where $k \in \{8, 16, 32\}$), the training set includes additional k cross-domain examples, which provide composition symbols, in the evaluation.

A.2. Experimental Setup for Compositionality

We include all the few-shot examples in the context to provide compositional symbols, and we retrieve single-domain exemplars with different retrievers. We omit the evaluation on 16-C and 32-C splits for the GPT-Neo model

Table 8. Inference latency on SST-5 validation set and evaluation metrics on different datasets when varying n at inference time.

Model	Latency	SST5	MRPC	QNLI	GeoQuery	NL2Bash	MTOP	Avg.
TOPK-BERT	30s	37.24	69.36	64.65	66.79	51.30	52.13	56.91
EPR	30s	42.82	75.98	80.76	68.57	56.82	64.20	64.86
CEIL (n=50)	30s	45.78	81.37	84.37	71.79	57.84	66.62	67.96
CEIL (n=100)	36s	47.05	80.15	85.41	**73.21**	59.91	67.43	68.86
CEIL (n=200)	55s	46.59	80.88	85.21	**73.21**	60.26	67.15	68.88
CEIL (n=400)	87s	47.14	**82.11**	85.46	72.86	**60.59**	**67.52**	69.28
CEIL (n=800)	118s	**47.32**	81.86	**86.21**	72.86	60.26	67.43	**69.32**

as we have no extra room due to the restriction of the input length. On Codex, we limit the number of in-context examples to 16 to fairly compare results across the different k-C splits.

B. Additional Experiments

B.1. Varying n at Inference Time

As discussed in §3.3 that we arrow down the candidate space with KNN retriever at inference time, we further conducted experiments on multiple datasets to investigate the effect of varying n. We show the inference latency on SST-5 validation set and evaluation metrics on different datasets in Table 8. Overall, we found that increasing n tends to improve performance, indicating that increasing n provides a larger exploration space and a higher chance of finding a better subset. In addition, inference efficiency is also an important consideration. The latency on the SST5 validation set demonstrates that increasing n will add extra overhead due to the complexity of the MAP inference algorithm, which results in a trade-off between performance and efficiency.

Furthermore, the impact of n on performance tends to become smaller as n increases. We show the distribution of the samples selected in the MAP subset from the top 800 candidate samples in Figure 4. Since both relevance and diversity are considered but relevance tends to have greater weight, the impact of n on performance diminishes because examples beyond the top 200 are not typically selected on most datasets. Therefore, although theoretically, a larger n will have a greater chance of finding a subset, from the perspective of the performance-efficiency trade-off and the diminishing returns of increasing n, we adopted an approximate approach that chooses a moderate amount of n.

B.2. Number of In-context Examples

We show additional results on the effect of in-context examples in Figure 5. We find CEIL mostly outperforms EPR and TOPK-BERT with 32 in-context examples by using merely 4 and 1 example, respective, greatly cutting down the computations as the attention module (Vaswani et al., 2017) in most LMs is of quadratic complexity.

C. Limitation

The main limitation of CEIL is inherent in the learning-based approach, which performs significantly better than learning-free methods but requires a certain amount of data to train the retriever for each task. The scoring stage in dataset construction of CEIL is also slower than EPR since we have to put an in-context example subset into the context instead of a single example. Although we have explored the transferability of the retriever, this research is still in its early stages. One potential avenue for future research is to use multitask-tuning to train a unified retriever so that the retriever can be applied directly to new tasks like in the learning-free approaches, without the need to retrain the retriever with new task data.

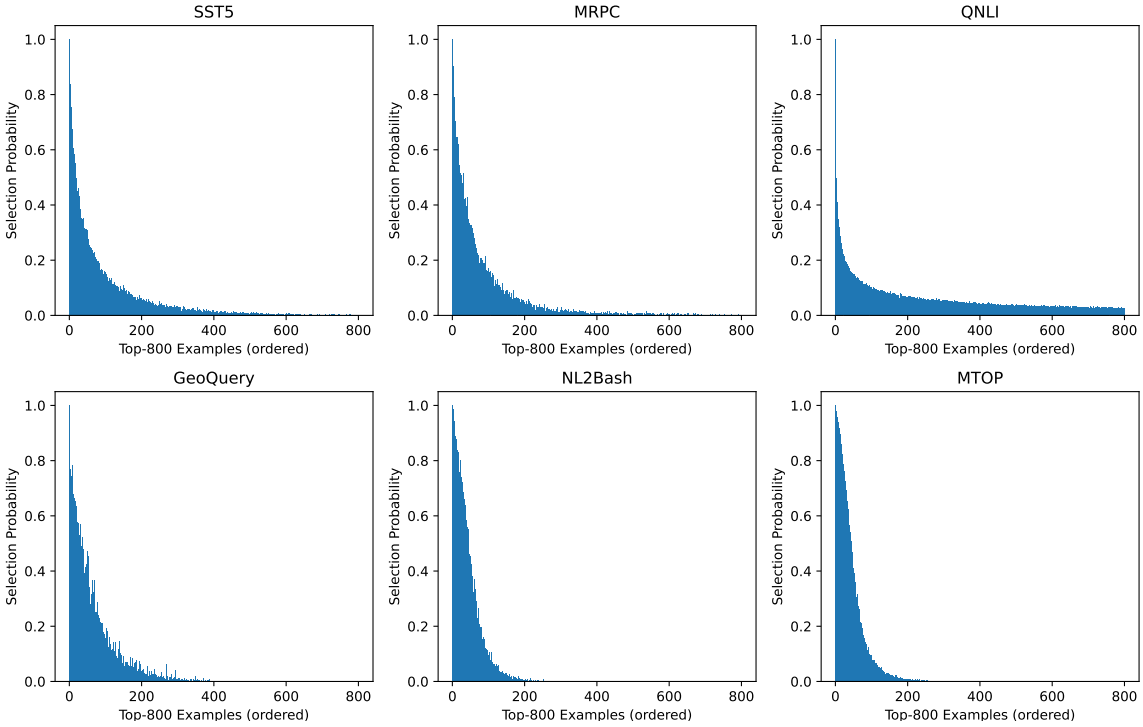

Figure 4. Distribution of the selection probability of the top-800 examples. As n increases, its impact on performance diminishes because examples beyond the top 200 are not typically selected on most datasets.

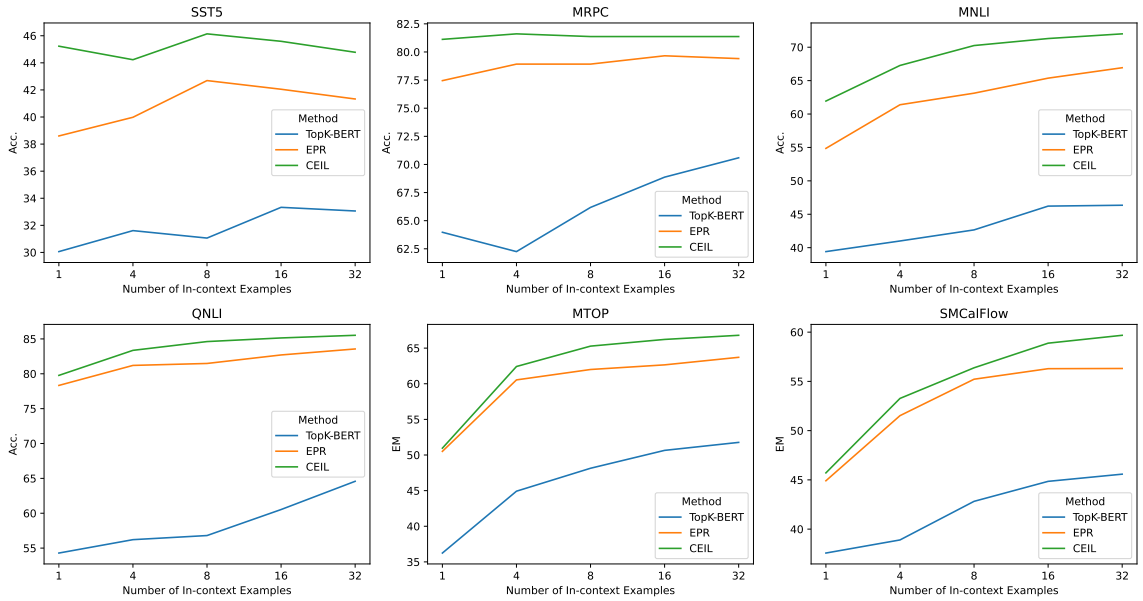

Figure 5. Comparison with baselines under various numbers of in-context examples.