Large Language Models Are Implicitly Topic Models: Explaining and Finding Good Demonstrations for In-Context Learning

Xinyi Wang [1] **Wanrong Zhu** [1] **Michael Saxon** [1] **Mark Steyvers** [2] **William Yang Wang** [1]

Abstract

In recent years, pre-trained large language models (LLMs) have demonstrated remarkable efficiency in achieving an inference-time few-shot learning capability known as in-context learning. However, existing literature has highlighted the sensitivity of this capability to the selection of few-shot demonstrations. Current understandings of the underlying mechanisms by which this capability arises from regular language model pretraining objectives remain disconnected from the real-world LLMs. This study aims to examine the in-context learning phenomenon through a Bayesian lens, viewing real-world LLMs as implicit topic models. On this premise, we propose an algorithm to select optimal demonstrations from a set of annotated data with a small LLM, then directly generalize the selected demonstrations to larger LLMs. We demonstrate a significant 12.5% improvement relative to the random selection baseline, averaged over eight GPT models on eight real-world text classification datasets. Our empirical findings support our hypothesis that LLMs implicitly infer a latent variable containing task information.

1. Introduction

Transformer-based (Vaswani et al., 2017) pre-trained large language models (LLMs) have demonstrated significant advancements in a variety of natural language processing (NLP) tasks. As the size of these LLMs increases, they gain "in-context learning" capabilities, whereby the models achieve state-of-the-art (SOTA) or near-SOTA performance by conditioning on a small number of demonstration examples at inference time, without any need for updating model parameters (Brown et al., 2020). Below is an example input sequence for semantic analysis with in-context learning:

```
Great movie.  Positive.

The worst movie ever.  Negative.

Can't wait to see the second movie!
```

The first two lines are two demonstrations, and the third line is a test input. We expect an LLM to output the correct label `Positive` as a continuation.

In-context learning has been demonstrated to be an effective technique for a wide range of NLP tasks. However, it is sensitive to the choice, format, and even the order of the demonstrations used (Perez et al., 2021; Lu et al., 2022). This makes achieving optimal performance with in-context learning a significant challenge, requiring real human effort to adjust the format and selection of demonstration examples. Heuristic solutions, such as selecting demonstrations based on the similarity between the demonstrations and test input (Liu et al., 2022; Su et al., 2022) have been proposed, but a comprehensive understanding of why certain demonstrations are effective while others are not remains elusive. Additionally, the mechanisms by which LLMs acquire in-context learning capabilities through training on natural text under the standard language model pre-training objective are not fully understood. Recent works on understanding in-context learning provide valuable insights and theoretical results (Chan et al., 2022; Akyürek et al., 2022; von Oswald et al., 2022; Jiang, 2023; Hahn & Goyal, 2023), but are limited in scope, focusing on synthetic experiments to validate their hypotheses, while it remains unclear if these results generalize to LLMs pre-trained on real-world natural language data. Xie et al. (2022) introduced a prominent result providing a latent topic (concept) variable interpretation for in-context learning. They showed that the in-context learning predictor approaches the Bayes optimal predictor when the number of demonstrations approaches infinity, under the assumption that both the pre-training data distribution and task-specific data distribution are Hidden Markov Models (HMM). However, the assumption that the data generation process is Hidden Markovian makes extrapolation of the result to natural language questionable, and restricts empirical verification to synthetic data with toy models.

We are inspired by this prior work and introduce a more general and natural explanation built on realistic assump-

[*]Equal contribution [1]Department of Computer Science, University of California, Santa Barbara [2]Department of Cognitive Sciences, University of California, Irvine. Correspondence to: Xinyi Wang <xinyi$_{w}ang@ucsb.edu$>.

Work presented at the ES-FoMo Workshop at ICML 2023.

tions, which gives rise to a practical demonstration selection algorithm. Our explanation is inspired by the generation process of a generic topic model:

$$P(\boldsymbol{w}_{1:T}) = \int_{\Theta} P(\boldsymbol{w}_{1:T}|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Where $\boldsymbol{\theta} \in \Theta$ represents a potentially high dimensional topic/concept variable, Θ is the space of the topic/concept variable, and $\boldsymbol{w}_{1:T}$ refers to the token sequence of a piece of text. On the other hand, generative LLMs model text data according to the general probabilistic decomposition:

$$P(\boldsymbol{w}_{1:T}) = \prod_{i=1}^{T} P(\boldsymbol{w}_i|\boldsymbol{w}_{i-1}, ..., \boldsymbol{w}_1)$$

While in practice, LLMs generate new tokens based on all previous tokens, we investigate whether a simplified assumption similar to that of topic models can be made for LLMs:

$$P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{w}_{1:t}) = \int_{\Theta} P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{\theta})P_M(\boldsymbol{\theta}|\boldsymbol{w}_{1:t})d\boldsymbol{\theta}$$

In this scenario, the generated tokens are assumed to be conditionally independent of previous tokens, given the latent topic (concept) variable that acts like an approximate sufficient statistic for the posterior information related to the prompt $\boldsymbol{w}_{1:t}$. For in-context learning, this concept variable includes format and task information. By conditioning on an appropriate latent concept variable, LLMs would generate the desired continuation with $P(\boldsymbol{w}_{t+1:T}|\boldsymbol{\theta})$. As LLMs do not explicitly learn a latent variable distribution like LDA-style topic models (Blei et al., 2003), we can instead utilize this formulation under an Empirical Bayesian formulation inspired by Lester et al. (2021) to only approximate the optimal latent variable value for a desired task, using a small LLM (with less than 1B parameters), which is computationally efficient.

We empirically validate our explanation by selecting examples ($\boldsymbol{w}_{1:t}$ in the equations) that are most likely to infer the optimal latent variable value (those with the highest posterior probability $P(\boldsymbol{\theta}|\boldsymbol{w}_{t+1:T})$). We then directly use them as demonstrations for in-context learning with other larger LLMs (up to 175B parameters) and observed a significant performance improvement. The generalization of demonstrations between LLMs is likely a result of similar pre-training data distributions.

While our work is inspired by that of Xie et al. (2022), our approach differs significantly in both theoretical analysis and experimental settings. Our main contributions are as follows:

- **We assume a general data generation process** specified by a three-variable causal graph, without constraints on the distribution function or the number of demonstrations.
- **We prove under these realistic assumptions** that the in-context learning predictor can reach the Bayes optimal predictor with a finite number of demonstrations chosen using the latent concept variable.
- **We introduce an efficient, practical demonstration selection algorithm** based on our theoretical results, which can select demonstrations using a small LLM and then directly generalize the demonstrations to other LLMs. The effectiveness of our algorithm is empirically validated using real-world LLMs and text classification tasks.

Our goal is to close the gap between theoretical understandings and real-world LLMs. To the best of our knowledge, our proposed latent variable explanation of in-context learning is the first Bayesian explanation that yields an effective algorithm in real-world scenarios.

2. Theoretical Analysis

In in-context learning, the prompt $w_{1:t}$ is composed of several demonstrations and a test input. The generated tokens $w_{t+1:T}$ represent the model's prediction for the test input.

2.1. Notations and Problem Setting

Suppose the objective of our task is to predict a discrete target variable $Y \in \mathcal{Y}$, given a token sequence $X \in \mathcal{X}$, where $\mathcal{X}$ is the space of all possible token sequences. $\boldsymbol{\theta} \in \Theta$ is a potentially high dimensional latent variable, where Θ is the high dimensional space of the variable. Unlike the traditional topic model, $\boldsymbol{\theta}$ is not assumed to be discrete, but continuously distributed over Θ. To define the data generation process, we posit the existence of an underlying causal relation between X, Y, and $\boldsymbol{\theta}$. We examine two potential directions of this causal relation, namely $X \to Y \leftarrow \boldsymbol{\theta}$ and $Y \to X \leftarrow \boldsymbol{\theta}$, which can be represented mathematically as the following structural equations:

$$Y = f(X, \boldsymbol{\theta}, \boldsymbol{\epsilon}) \qquad X = g(Y, \boldsymbol{\theta}, \boldsymbol{\epsilon})$$

Here $\boldsymbol{\epsilon} \in \mathcal{E}$ is an independent noise variable, $f : \mathcal{X} \times \Theta \times \mathcal{E} \to \mathcal{Y}$ and $g : \mathcal{Y} \times \Theta \times \mathcal{E} \to \mathcal{X}$ are two deterministic functions. Furthermore, we denote the joint data distribution by $X, Y, \boldsymbol{\theta} \sim P$, and assume that Y is sampled from a uniform distribution over $\mathcal{Y}$. The distinction between these two directions is crucial, as it allows us to utilize the direction in which the child variable (Y or X) is independent of the other variables, given its parents.

We hypothesize that the causal direction depends on the nature of the task. For instance, in the task of predicting the sentiment (Y) of a movie review (X), it is reasonable to assume that the opinion about the movie is formed before writing the review, thus making Y the cause of X, along with the task concept of "writing a passage to express one's opinion about the movie" ($\boldsymbol{\theta}$). Conversely, for the task of classifying whether a product review (X) is helpful to other customers (Y), it is the quality of the review (X) that cause other customers to upvote it (Y), along with the task concept of "rating the helpfulness of this review" ($\boldsymbol{\theta}$). *In the rest of the paper, we will focus on the $X \rightarrow Y \leftarrow \boldsymbol{\theta}$ direction and leave a detailed discussion of the other direction in the Appendix.*

Suppose we are interested in a task (e.g. semantic analysis) denoted by $d \in \mathcal{T}$, where $\mathcal{T}$ is the space of all possible tasks. We assume there is an injective function between $\mathcal{T}$ and Θ. i.e. for each task d, there is a concept variable θ^d, such that each data (X^d, Y^d) sampled from task d is generated by:

$$Y^d = f(X^d, \theta^d, \boldsymbol{\epsilon})$$

To perform in-context learning with an LLM (generically denoted by model label M), we condition on a fixed set of k demonstration examples $(X_1^d, Y_1^d), (X_2^d, Y_2^d), ..., (X_k^d, Y_k^d)$ sampled from task d.

Following previous works (Min et al., 2022a;c), as we are not using any instruction fine-tuned models, we do not include a task description in the prompt, with the aim of focusing on the examination of the demonstrations. To naturally project $\mathcal{Y}$ into the token space $\mathcal{X}$, we define injective mappings $\tau^d : \mathcal{Y} \rightarrow \mathcal{X}$, which are typically defined by human understanding of the task d. e.g. for sentiment analysis, τ^d map positive class to the token "positive" and negative class to the token "negative". Additionally, a delimiter token $\boldsymbol{w}^d$ is defined, typically an empty space or a new line token, to separate the demonstrations when concatenated. We denote the LLM output probability of X, Y, and $\boldsymbol{\theta}$, with the aforementioned preprocessing applied, by P_M^d:

$$\begin{aligned}&P_M(\tau^d(Y)|X_1^d, \tau^d(Y_1^d), \boldsymbol{w}^d, ..., X_k^d, \tau^d(Y_k^d), \boldsymbol{w}^d, X)\\ =&P_M^d(Y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)\end{aligned}$$

2.2. Problem Analysis and Theoretical Results

Suppose a set of observed data sampled from task d, denoted as $\mathcal{D}^d$, is available, allowing for the selection of the k most suitable demonstrations from it. For any incoming test example X, we have:

$$\begin{aligned}&P_M^d(Y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)\\ =&\int_\Theta P_M^d(Y|\boldsymbol{\theta}, X) P_M^d(\boldsymbol{\theta}|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X) d\boldsymbol{\theta}\end{aligned} \tag{1}$$

Here, we assume the sampling of the test example is independent of the sampling of the demonstrations, so Y is independent of the demonstrations given $\boldsymbol{\theta}$ and X. We also assume that the pre-trained data distribution P_M^d is a suitable approximation of the assumed data distribution P:

Assumption 2.1. Assume that $P_M(X) = P(X)$, and $P_M^d(Y|\boldsymbol{\theta}, X) \propto P(Y|\boldsymbol{\theta}, X)$ for $X \rightarrow Y \leftarrow \boldsymbol{\theta}$.

Note that the assumption that a large language model captures the true distribution of language is fairly common in the literature studying LLMs (Xie et al., 2022; Saunshi et al., 2021; Wei et al., 2021). With this assumption, we establish:

Proposition 2.2. *If task d follows the $X \rightarrow Y \leftarrow \boldsymbol{\theta}$ direction, then $\arg\max_{y \in \mathcal{Y}} P_M^d(Y = y|\theta^d, X)$ is the Bayes optimal classifier.*

In this case, only when $P_M^d(\boldsymbol{\theta}|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$ completely concentrate on θ^d, can the in-context learning classifier become the Bayes optimal classifier (Devroye et al., 1996):

Theorem 2.3. *If task d follows the $X \rightarrow Y \leftarrow \boldsymbol{\theta}$ direction, then the in-context learning classifier*

$$\arg\max_{y \in \mathcal{Y}} P_M^d(Y = y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$$

always has a higher or equal probability of misclassification to the Bayes optimal classifier $\arg\max_{y \in \mathcal{Y}} P_M^d(Y = y|\theta^d, X)$. Equality only holds when

$$\forall x \in \mathcal{X},\ P_M^d(\theta^d|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X = x) = 1.$$

A similar argument can be made for the $Y \rightarrow X \leftarrow \boldsymbol{\theta}$ direction. [1] Here, Equation (1) would become:

$$\begin{aligned}&P_M^d(X|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y)\\ =&\int_\Theta P_M^d(X|\boldsymbol{\theta}, Y) P_M^d(\boldsymbol{\theta}|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y) d\boldsymbol{\theta}\end{aligned} \tag{2}$$

Note that the left-hand side of Equation (1) and Equation (2) are similar to the direct and channel method introduced by Min et al. (2022a). However, our analysis differs from theirs in that we do not treat ($Y \rightarrow X \leftarrow \boldsymbol{\theta}$) as the universally superior channel direction for modeling in-context learning, rather arguing that depending on the end task, the causal direction ($X \rightarrow Y \leftarrow \boldsymbol{\theta}$) is sometimes better. This view is supported by our empirical results in Appendix B.

3. Method

Here we demonstrate how the proposed theory can be practically applied to select optimal demonstration examples. Since latent variable $\boldsymbol{\theta}$ encodes both the task and format information, the whole distribution over Θ is too complex to model. Unlike traditional topic models, we will only focus on estimating an optimal value θ^d corresponding to task d.

[1]The detailed argument of the $Y \rightarrow X \leftarrow \boldsymbol{\theta}$ direction can be found in Appendix A.2.

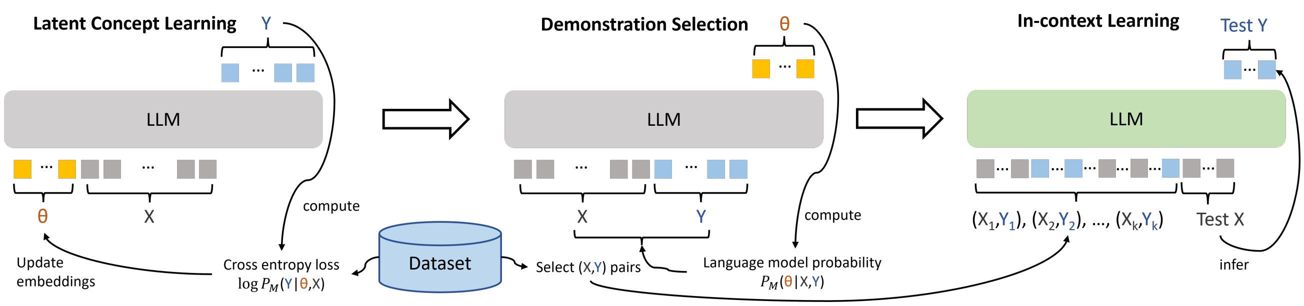

Figure 1. An overview of our proposed two-phased algorithm. Demonstration selection and latent concept learning share the same LLM as demonstration selection needs to reuse the learned concept tokens, while at the in-context learning time, any other generative LLMs can be used. Here we only illustrate the $X \rightarrow Y \leftarrow \boldsymbol{\theta}$ direction. The $Y \rightarrow X \leftarrow \boldsymbol{\theta}$ direction can be illustrated similarly by exchanging X and Y in the above figure.

Algorithm 1 Latent concept learning

Input: Dataset $\mathcal{D} = \{(x_i, y_i, d_i)\}_i$ associated with a set of tasks $\mathcal{S}$, LLM M, number of concept tokens per task c, learning rate α, and number of training steps N.
Output: LLM M' with fine-tuned concept tokens.
Add $c|\mathcal{S}|$ new tokens to the vocabulary. i.e. The concept tokens $\hat{\theta}^d$ for each task in $\mathcal{S}$. Randomly initialize their embeddings E_{new}. Freeze all parameters in M except E_{new};
for step $= 1$ **to** N **do**
  Sample a random batch B in $\mathcal{D}$ and initialize gradient $g \leftarrow 0$;
  for each data point (x, y, d) in B **do**
    $g = g + \frac{\partial \ell(X,Y;\hat{\theta}^d)}{\partial E_{new}}$;
  end for
  $E_{new} = E_{new} - \alpha g$;
end for

First, we perform *latent concept learning*, wherein the task latent θ^d is learned as a set of new token embeddings using prompt tuning over the full demonstration candidate set. With this optimal task latent, we then perform *demonstration selection*, where a smaller set of demonstrations is chosen to maximize the likelihood of postfixing the latent concept tokens. We only need to use a small LLM to do the above steps to obtain an optimal set of demonstrations that can be directly transferred to other LLMs. Figure 1 is an overall illustration of our proposed method.

3.1. Latent Concept Learning

We want to first find the optimal value of the latent concept variable θ^d corresponding to a task $d \in \mathcal{T}$. As $\arg\max_{y \in \mathcal{Y}} P_M^d(Y = y|\theta^d, X)$ is the Bayes optimal classifier according to Proposition 2.2, θ^d should be able to minimize $-\mathbb{E}_{X,Y,d}[\log P_M^d(Y|\theta^d, X)]$ for the $X \rightarrow Y \leftarrow \boldsymbol{\theta}$ direction. In practice, we try to align θ^d to the token embedding space by adding new tokens to the vocabulary. After this alignment, we hope to be able to use the learned new tokens of θ^d as regular tokens.

More specifically, building upon the methodology proposed by Lester et al. (2021), for each specific task d, c new concept tokens (denoted as $\hat{\theta}^d$) are added to the original vocabulary of LLM M to represent the corresponding task concept θ^d. Subsequently, the embedding of these new tokens $E_{new}(\hat{\theta}^d)$ is fine-tuned while freezing the remaining parameters of LLM M. The variable c is treated as a hyperparameter. In practice, in order to condition on θ^d, the corresponding c concept tokens are appended to the input X (or Y) as shown in the example provided below, where $c = 2$:

```
<sentiment_token_1><sentiment_token_2>
Can't wait to see the second movie!
```

By giving the above input tokens, we ask the LLM to predict the correct label `Positive` for us. Note that <sentiment_token_1> here is just a label assigned to the newly added concept token. It can be anything as long as it does not overlap with the original vocabulary of LLM.

The fine-tuning objective would then be minimizing $\mathcal{L}(\hat{\theta}^d) = \mathbb{E}_{X,Y}[\ell(X, Y; \hat{\theta}^d)]$, where

$$\ell(X, Y; \hat{\theta}^d) = \begin{cases} -\log P_M^d(Y|\hat{\theta}^d, X) & \text{if } X \rightarrow Y \leftarrow \boldsymbol{\theta} \\ -\log P_M^d(X|\hat{\theta}^d, Y) & \text{if } Y \rightarrow X \leftarrow \boldsymbol{\theta}. \end{cases}$$

Theoretically, if we can minimize the above loss function, a Bayes optimal classifier can be obtained, and the concept tokens would be a reasonable delegate of the real latent concept variable:

Proposition 3.1. *When $\mathcal{L}(\hat{\theta}^d)$ is minimized, $P_M^d(Y|\hat{\theta}^d, X) = P(Y|\theta^d, X)$ for $X \rightarrow Y \leftarrow \boldsymbol{\theta}$. If the LLM M is invertible, then $\hat{\theta}^d = \theta^d$.*[2]

[2]More discussion can be found in Appendix A.3.

We denote the LLM M with fine-tuned concept tokens by M'. Since we add the concept tokens into the regular token vocabulary, the raw LLM output probability $P_{M'}(\hat{\theta}^d|\boldsymbol{w}_{1:t})$ ($\boldsymbol{w}_{1:t}$ denote a given prompt) would be in the token sequence space $\mathcal{X}$ instead of the concept space Θ. Since learning all possible $\theta^d \in \Theta$ is infeasible, we propose to approximate the concept space Θ by sampling a diverse subset of tasks $\mathcal{S} \subseteq \mathcal{T}$. Then the estimated conditional probability of θ^d would be:

$$\hat{P}^d_{M'}(\hat{\theta}^d|\boldsymbol{w}_{1:t}) = \frac{P^d_{M'}(\hat{\theta}^d|\boldsymbol{w}_{1:t})}{\sum_{t\in\mathcal{S}} P^t_{M'}(\hat{\theta}^t|\boldsymbol{w}_{1:t})}$$

To obtain the concept tokens for all tasks in $\mathcal{S}$, we fine-tune all tasks together with the loss $\sum_{d\in\mathcal{S}} \mathcal{L}(\theta^d)$. We summarize the proposed algorithm in Algorithm 1.

Note that the embedding matrix of a generative LLM is shared on both the input and output sides. So while we only see the concept tokens on the input side at the training time, they can be viewed as regular word tokens that can be generated on the output side.

3.2. Demonstration Selection

According to Theorem 2.3, for a task d, to make the in-context learning classifier closer to the Bayes optimal classifier, we need to select demonstrations $(X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)$ that maximize $P^d_M(\theta^d|X^d_1, Y^d_1, ..., X^d_k, Y^d_k, X)$ for all $X \in \mathcal{X}$. Then our goal then becomes selecting demonstrations that can best infer the task concept for all test inputs on average:

$$\arg\max_{X^d_1, Y^d_1, ..., X^d_k, Y^d_k} \mathbb{E}_X[P^d_M(\theta^d|X^d_1, Y^d_1, ..., X^d_k, Y^d_k, X)]$$

As test examples are sampled independent of the demonstrations, and $P_M(X) = P(X)$ according to Assumption 2.1, we have

$$\begin{aligned}&\mathbb{E}_X[P^d_M(\theta^d|X^d_1, Y^d_1, ..., X^d_k, Y^d_k, X)]\\ =&P^d_M(\theta^d|X^d_1, Y^d_1, ..., X^d_k, Y^d_k)\end{aligned}$$

If we assume each demonstration is also sampled independently, we have:

$$P^d_M(\theta^d|X^d_1, Y^d_1, ..., X^d_k, Y^d_k) = \frac{\prod_{i=1}^k P^d_M(\theta^d|X^d_i, Y^d_i)}{P^d_M(\theta^d)^{k-1}}$$

Assuming that $\boldsymbol{\theta}$ has a uniform prior, then our goal becomes finding the top k demonstrations that maximize $\hat{P}^d_{M'}(\hat{\theta}^d|X^d_i, Y^d_i)$. Note that the independence between demonstrations is a simplified assumption to reduce the combinatory search space of $(X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)$. In practice, selected demonstrations are likely correlated as some demonstrations may work well together but not necessarily work well by themselves. However, it would be too expensive to search the $O(|\mathcal{D}^d|^k)$ combinations over the candidate set $\mathcal{D}^d$. In practice, this simplification works reasonably well. We leave this combinatory search problem to future research.

Algorithm 2 Demonstration selection

Input: dataset $\mathcal{D}^d$ for a task d. LLM with fine-tuned concept tokens M'. The number of demonstrations k.
Output: A set of ordered demonstrations.
for each (X^d, Y^d) in $\mathcal{D}^d$ **do**
  Compute $\hat{P}^d_M(\hat{\theta}^d|X^d, Y^d)$;
end for
Select top k examples with the largest $\hat{P}^d_M(\hat{\theta}^d|X^d, Y^d)$, denoted as $(X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)$;
for each permutation π **do**
  Compute $\hat{P}^d_M(\hat{\theta}^d|\pi((X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)))$;
end for
Select the permutation π with the largest $\hat{P}^d_M(\hat{\theta}^d|\pi((X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)))$.

Also, as we are using an LLM to approximate the data distribution, the order of the demonstrations likely matters. We then choose the order according to the posterior of the concept tokens:

$$\arg\max_{\pi\in\Pi} \hat{P}^d_M(\theta^d|\pi((X^d_1, Y^d_1), ..., (X^d_k, Y^d_k))) \quad (3)$$

Where $\pi((X^d_1, Y^d_1), ..., (X^d_k, Y^d_k))$ is a permutation of $(X^d_1, Y^d_1), ..., (X^d_k, Y^d_k)$. Π is the set of all possible permutations of the k demonstrations. We summarize the proposed algorithm in Algorithm 2.

4. Experiments

Datasets. We conduct experiments on eight datasets from five different types of NLP classification tasks: sentiment analysis, linguistic analysis, topic classification, emotion classification, and hate speech detection. For sentiment analysis, we choose the Stanford Sentiment Treebank (SST2) dataset (Socher et al., 2013) from the GLUE benchmark (Wang et al., 2018) and the financial phrase bank (FPB) dataset (Malo et al., 2014). SST2 is constructed based on movie reviews labeled "positive" or "negative", and FPB is based on financial news labeled "positive", "negative", or "neutral". For linguistic analysis, we choose the Corpus of Linguistic Acceptability (COLA) dataset (Warstadt et al., 2018) from the GLUE benchmark, based on sentences collected from linguistic books, labeled with "acceptable" or "unacceptable". For topic classification, we choose the DBpedia ontology classification dataset (Zhang et al., 2015), based on DBpedia 2014 (Lehmann et al., 2015), labeled with 14 different ontology classes. For emotion classification, we choose the dataset from Chatterjee et al. (2019) and

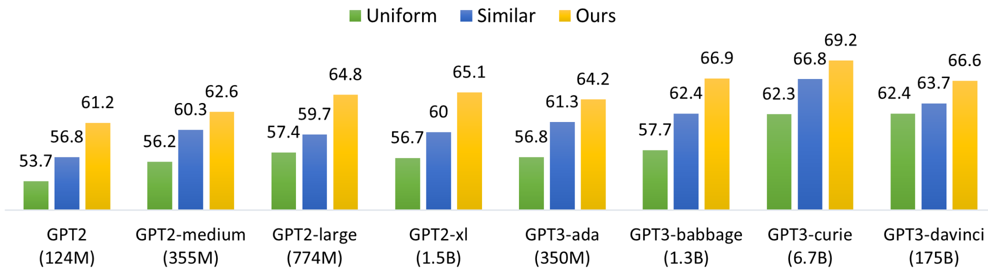

Figure 2. Accuracy of 4-shot in-context learning using demonstrations selected by our method and other baselines, averaged over eight datasets. Our demonstrations are selected using GPT2-large, and the same set of demonstrations is then applied to all other LLMs.

Saravia et al. (2018), both of which are collected from Twitter. Chatterjee et al. (2019) (EmoC) predict emotion given a three-turn contextual dialogue, while Saravia et al. (2018) predict emotion given a Twitter message with clear emotion. For hate speech detection, we choose the online hate speech detection dataset (ETHOS) (Mollas et al., 2020), collected from online social media platforms. Here we detect two types of hate speech: sexual orientation (ETHOS-SO) and religion (ETHOS-R). While in Section 2, we assume that all tasks share the same label space $\mathcal{Y}$, here we relax such assumption and allow a different number of labels for different tasks. We use minimal formatting to process each example. A detailed description of the datasets and our data processing procedure can be found in Appendix B.

Experiment settings. To determine the causal direction for each task, we select the direction that can give higher accuracy when using random demonstrations[3]. We adopt the $Y \to X \leftarrow \boldsymbol{\theta}$ direction for sentiment analysis, topic classification, and emotion classification tasks, which is consistent with the intuition that people usually have some sentiment, topic, or emotion in mind before writing a piece of text. We adopt the $X \to Y \leftarrow \boldsymbol{\theta}$ direction for the linguistic analysis and hate speech detection type of tasks. While this is less intuitive, we can understand this as linguistic error and hate speech detection are more of a post hoc task in contrast to the previous tasks.

Without specification, we use $k = 4$ number of demonstrations and $c = 10$ number of concept tokens per dataset for our experiments, as the context length of GPT2 is 1024, and a larger number of demonstrations may not be able to completely fit into it. We use GPT2-large to learn the concept tokens and then compute the probability of each candidate demonstration example. We select our demonstrations from a randomly selected 100 example subset of the train set as the candidate set $\mathcal{D}^d$. We use the same set of demonstrations selected by GPT2-large for all other LLMs. We test the performance of the selected demonstrations using at most 1000 examples randomly sampled from the test set. Each experiment is repeated for five runs with different random seeds (the randomness comes from the sampling of the candidate set and the sampling of the test set). We adopt a large portion of the code from Min et al. (2022b), which is based on Huggingface (Wolf et al., 2019).

Baselines. We consider the following baselines:

- **Uniform**: We uniformly select k demonstrations from $\mathcal{D}$ for each test example.
- **Similar**: According to Liu et al. (2022), demonstrations that are semantically similar to the test example would hare more performant. Following their method, we use a pre-trained sentence Transformer (Reimers & Gurevych, 2019) to calculate the cosine similarity between the demonstrations and test examples. We choose the top k similar demonstrations from $\mathcal{D}$ for each test example.

Main results.[4] Figure 2 shows our main results averaged over all eight datasets, using the first-generation GPT2s and GPT3s, without any instruction fine-tuning (Ouyang et al., 2022) or Reinforcement Learning from Human Feedback (RLHF) (Stiennon et al., 2020). Our method significantly outperforms baselines on eight different LLMs, with 12.5% relative improvement to the uniform selection baseline on average, which shows the effectiveness of our method. The demonstrations selected by our method are exclusively based on GPT2-large, while the same set of demonstrations can be generalized to all other GPTs.

Results with non-GPT models. In Figure 3, we test the demonstrations selected by our method using GPT2-large on more LLMs (GPT3 (Brown et al., 2020), GPT3-instruct (Ouyang et al., 2022; Stiennon et al., 2020), GPT-J (Wang & Komatsuzaki, 2021), OPT (Zhang et al., 2022), and LLaMA (Touvron et al., 2023)) with similar sizes (6-7B), and show that the selected demonstrations improve in-context learning performance of all of them. The fact that GPT3-curie obtains the largest performance improvement is likely because similar pre-training data distributions help the generalization

[3]Detailed results see Figure 8 in Appendix B.

[4]The complete results with standard deviations in this section can be found in Appendix B.

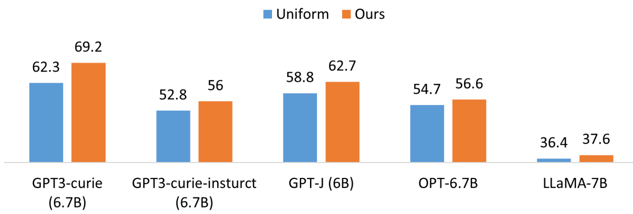

Figure 3. Proposed method v.s. randomly selected demonstrations. In-context learning accuracy averaged over all eight datasets.

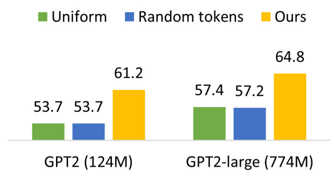

Figure 4. Proposed method v.s. using randomly selected tokens. In-context learning accuracy averaged over all eight datasets.

of the selected demonstrations. Different-size GPT2 models share the same pre-training corpus (Radford et al., 2019), while GPT3s are pre-trained on a dataset expanded from the GPT2 pre-training corpus (Brown et al., 2020). Thus the pre-training distribution of GPT3-curie and GPT2-large can be assumed to be similar.

Learned tokens v.s. Random tokens. To confirm the critical role of the latent concept variable in the proposed demonstration selection algorithm, we compare the performance of using the learned concept tokens versus using randomly selected tokens from the original vocabulary in Figure 4. The demonstrations selected by random tokens only obtain the same performance as randomly selected demonstrations, showing that the performance gain of our method comes from the learned concept tokens containing the task and format information, not other elements of our algorithm.

k ablation study. While we use $k = 4$ demonstrations for all experiments, we also test the effectiveness of our method using different k. As shown in Figure 5, our method significantly outperforms the random selection baseline with $k = 2$, 4, 8, and 16. To fit in large ks, we use GPT3-ada with a longer context length (2048). Note that for real-world tasks, it is in general not true that more demonstrations guarantee higher performance (Chen et al., 2023). We can

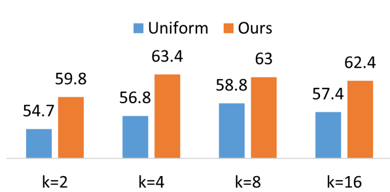

Figure 5. k ablation study. In-context learning accuracy of our method versus random selection baseline averaged over all eight datasets with GPT3-ada.

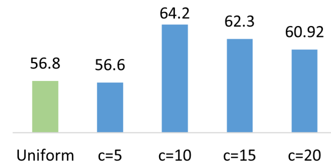

Figure 6. c ablation study. In-context learning accuracy of our method versus random selection baseline averaged over all eight datasets with GPT3-ada.

see that the uniform baseline performance increases from $k = 2$ to $k = 8$, then drops a little at $k = 16$. Our method improves the uniform baseline by around 5% absolute for all ks, while $k = 4$ improves the most (6.6%). Our method appears to have a diminishing effect when k becomes larger, which is likely because the effect of more demonstrations overwhelms the effect of demonstration choices.

c ablation study. While we use $c = 10$ number of concept tokens for all experiments, we also investigate the effect of different c on our method. When c is small ($c = 5$), the concept tokens cannot effectively capture the task and format information, thus cannot improve the performance. When c increases from 10 to 20, we observe a drop in the performance. It is likely because the selectivity of the concept tokens decreases when c increases. The longer the concept token sequence is, the more likely it will contain meaningless tokens that do not contribute to demonstration selection.

Effect of demonstrations' order. We find that the demonstrations selected by our method are insensitive to their order in most cases.[5] An exception is the EmoC dataset, where our method has a high variance. On the contrary, Lu et al. (2022) found that the order of the demonstration matters, and a good ordering cannot be transferred between different LLMs. We suspect that the ordering only matters when the demonstration selection method is not robust. Since Lu et al. (2022) randomly selects one set of demonstrations for the whole test set, the variance in performance is high with different demonstrations, thus ordering matters. And since such ordering is not transferable while our selected demonstrations are highly transferable, we suspect the core task information is stored in the content of the demonstrations, while the ordering mainly captures model-specific artifacts.

Qualitative illustration. In Figure 7, we provide a t-SNE (van der Maaten & Hinton, 2008) projection of the learned concept token embeddings. The tokens corresponding to semantically similar tasks are close together. Note that this result only aim to provide a straightforward illustration of concept tokens. The effect of concept tokens should be understood by the previous quantitative results.[6]

[5] Detailed results see Figure 11 in Appendix B.

[6] The list of similar tokens for these concept tokens can be found in Table 12 in Appendix B.

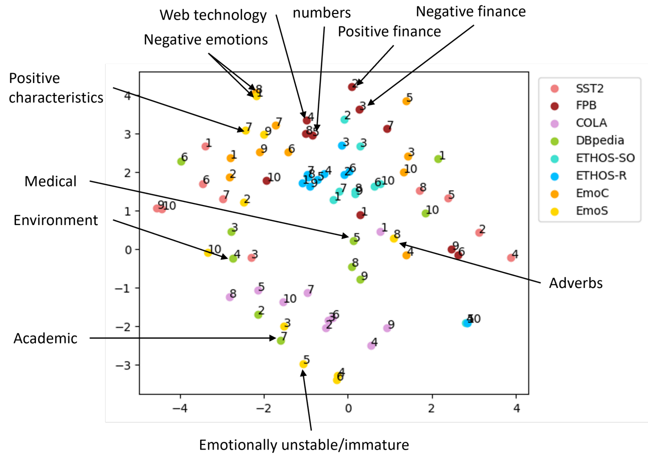

Figure 7. t-SNE plot of the learned concept tokens for each task. Concept tokens that can be explained by similar tokens are summarized in the graph.

5. Related Work

Heuristic solutions, such as selecting demonstrations based on the similarity between the demonstrations and test input (Liu et al., 2022; Su et al., 2022; Rubin et al., 2021) have been proposed. (Lu et al., 2022) propose to reorder the demonstration based on the entropy of the predicted labels. In this paper, we use the similarity-based selection method as a baseline while do not include the label entropy-based reordering method as we show that the ordering of the demonstrations does not matter for our method.

Previous research on the phenomenon of in-context learning in Transformers has identified a number of pre-training data distributions that can lead to the emergence of this capability, including a Hidden Markov Model distribution (Xie et al., 2022) and a skewed Zipfian distribution with high burstiness (Chan et al., 2022). Other studies have sought to understand the underlying mechanisms of in-context learning by making connections with gradient descent (von Oswald et al., 2022; Dai et al., 2022; Akyürek et al., 2022), formalizing it as an algorithm learning problem (Li et al., 2023), or proposing a latent variable theory similar as ours (Jiang, 2023; Hahn & Goyal, 2023; Xie et al., 2022). While providing valuable insights on how in-context learning works, these works are limited to synthetic datasets and toy Transformers, while it remains unclear if these results generalize to LLMs pre-trained on real-world text data and whether these results can help in-context learning performance. In contrast, we propose a Bayesian explanation of in-context learning that can be verified with real-world LLMs on various NLP datasets. Dai et al. (2022) provide a practical algorithm based on the understanding that the Transformer has a dual form of gradient descent. However, their empirical results are smaller in scale, with six datasets and only one model (350M), and has less significant improvements (5.4% relative to baseline).

There are also works trying to understand in-context learning from an empirical perspective (Bansal et al., 2022; Min et al., 2022a). Min et al. (2022c) found demonstrations' ground truth labels do not matter for in-context learning, which we find is not entirely accurate in Appendix B. On the other hand, chain-of-thoughts prompting (Wei et al., 2022; Zhou et al., 2022; Wang et al., 2022) find that providing step-by-step explanations improves in-context learning performance.

6. Conclusion

In this work, we endeavor to comprehend large language models (LLMs) through a Bayesian lens and posit them as implicit topic models that infer a latent conceptual variable from prompts. Motivated by this understanding, we propose a two-step algorithm that first extracts latent conceptual tokens from a small LLM and then selects demonstrations that have the greatest probability of predicting the corresponding conceptual tokens. The selected demonstrations can then be directly generalized to other LLMs. The efficacy of our algorithm across various text classification datasets and GPT models validates our explanation of in-context learning.

References

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Bansal, H., Gopalakrishnan, K., Dingliwal, S., Bodapati, S., Kirchhoff, K., and Roth, D. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. *arXiv preprint arXiv:2212.09095*, 2022.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003. ISSN 1532-4435.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chan, S. C., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A., Richemond, P. H., McClelland, J., and Hill, F. Data distributional properties drive emergent few-shot learning in transformers. *arXiv preprint arXiv:2205.05055*, 2022.

Chatterjee, A., Narahari, K. N., Joshi, M., and Agrawal, P. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 39–48, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2005. URL https://www.aclweb.org/anthology/S19-2005.

Chen, J., Chen, L., and Zhou, T. It takes one to tango but more make trouble? in-context training with different number of demonstrations. *arXiv preprint arXiv:2303.08119*, 2023.

Dai, D., Sun, Y., Dong, L., Hao, Y., Sui, Z., and Wei, F. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

Devroye, L., Györfi, L., and Lugosi, G. A probabilistic theory of pattern recognition. In *Stochastic Modelling and Applied Probability*, 1996.

Hahn, M. and Goyal, N. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Jiang, H. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

LeBrun, B., Sordoni, A., and O'Donnell, T. J. Evaluating distributional distortion in neural language modeling. In *International Conference on Learning Representations*, 2022.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. Transformers as algorithms: Generalization and implicit model selection in in-context learning. *arXiv preprint arXiv:2301.07067*, 2023.

Liu, J., Shen, D., Zhang, Y., Dolan, 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, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10.

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.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J., and Takala, P. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014.

Min, S., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5316–5330, 2022a.

Min, S., Lewis, M., Zettlemoyer, L., and Hajishirzi, H. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, Seattle, United States, July 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201. URL https://aclanthology.org/2022.naacl-main.201.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*, 2022c.

Mollas, I., Chrysopoulou, Z., Karlos, S., and Tsoumakas, G. Ethos: an online hate speech detection dataset, 2020.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Perez, E., Kiela, D., and Cho, K. True few-shot learning with language models. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=ShnM-rRh4T.

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

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Rubin, O., Herzig, J., and Berant, J. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL https://www.aclweb.org/anthology/D18-1404.

Saunshi, N., Malladi, S., and Arora, S. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=vVjIW3sEc1s.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., 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, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

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.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

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.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP 2018*, pp. 353, 2018.

Wang, B. and Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

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.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Wei, C., Xie, S. M., and Ma, T. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Neural Information Processing Systems (NeurIPS)*, 2021.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Bousquet, O., Le, Q., and Chi, E. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

A. Proofs

A.1. Direct direction

Assumption A.1. (Assumption 2.1) Assume that $P_M(X) = P(X)$, and $P_M^d(Y|\boldsymbol{\theta}, X) \propto P(Y|\boldsymbol{\theta}, X)$ for $X \to Y \leftarrow \boldsymbol{\theta}$.

Proposition A.2. *(Proposition 2.2) If task d follows the $X \to Y \leftarrow \boldsymbol{\theta}$ direction,* $\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\theta^d, X)$ *is the Bayes optimal classifier.*

Proof. Since the data generation of the task d can be written as $Y = f(X, \theta^d, \boldsymbol{\epsilon})$, we have

$$P^d(Y|X) = P(Y|\theta^d, X).$$

And by Assumption A.1, we have

$$\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\theta^d, X) = \arg\max_{y\in\mathcal{Y}} P(Y = y|\theta^d, X).$$

Thus $\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\theta^d, X)$ is the Bayes optimal classifier. □

Theorem A.3. *(Theorem 2.3) If task d follows the $X \to Y \leftarrow \boldsymbol{\theta}$ direction, then the in-context learning classifier*

$$\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$$

always has a higher or equal probability of misclassification to the Bayes optimal classifier $\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\theta^d, X)$. *Equality only takes when*

$$\forall x \in \mathcal{X},\ P_M^d(\theta^d|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X = x) = 1.$$

Proof. Recall that in Equation (1), we have

$$P_M^d(Y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X) = \int_\Theta P_M^d(Y|\boldsymbol{\theta}, X) P_M^d(\boldsymbol{\theta}|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X) d\boldsymbol{\theta}.$$

By Proposition A.2, $\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\theta^d, X)$ is the Bayes optimal classifier. Let $C_{\boldsymbol{\theta}}(X) = \arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|\boldsymbol{\theta}, X)$, then the risk is defined as the probability of misclassification

$$R(C_{\boldsymbol{\theta}}) = P(C_{\boldsymbol{\theta}}(X) \neq Y) = \mathbb{E}_{XY}[\mathbb{1}_{C_{\boldsymbol{\theta}}(X)\neq Y}].$$

Denote the in-context learning classifier $\arg\max_{y\in\mathcal{Y}} P_M^d(Y = y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$ by $C_k(X)$. We then have

$$R(C_k) = \mathbb{E}_{XY}[\mathbb{1}_{C_k(X)\neq Y}] = \mathbb{E}_X[\sum_{y\in\mathcal{Y}}(1 - P_M^d(Y = y|\theta^d, X))\mathbb{1}_{C_k(X)=y}].$$

Such risk is minimized if and only if $C_k(X) = C_{\theta^d}(X)$, which only holds when $P_M^d(\theta^d|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X = x) = 1$ for all $x \in \mathcal{X}$. □

A.2. Channel direction

Assumption A.4. Assume that $P_M(X) = P(X)$, and $P_M^d(X|\boldsymbol{\theta}, Y) \propto P(X|\boldsymbol{\theta}, Y)$ for the $Y \to X \leftarrow \boldsymbol{\theta}$ direction.

Proposition A.5. *If task d follows the $Y \to X \leftarrow \boldsymbol{\theta}$ causal direction,* $\arg\max_{y\in\mathcal{Y}} P_M^d(X|\theta^d, Y = y)$ *is the Bayes optimal classifier when the label assignment is balanced.*

Proof. Since the data generation of the task d can be written as $X = g(Y, \theta^d, \boldsymbol{\epsilon})$, we have

$$P^d(X|Y) = P(X|\theta^d, Y)$$

When the label is balanced, i.e. $P^d(Y) = \frac{1}{|\mathcal{Y}|}$, we have

$$P^d(Y|X) = \frac{P^d(X|Y)P^d(Y)}{P(X)} \propto P^d(X|Y)$$

And by Assumption A.4, we have

$$\arg\max_{y \in \mathcal{Y}} P_M^d(X|\theta^d, Y = y) = \arg\max_{y \in \mathcal{Y}} P(X|\theta^d, Y = y).$$

Thus $\arg\max_{y \in \mathcal{Y}} P_M^d(X|\theta^d, Y = y) = \arg\max_{y \in \mathcal{Y}} P^d(Y = y|X)$ is the Bayes optimal classifier. □

Theorem A.6. *If task d follows the $Y \to X \leftarrow \boldsymbol{\theta}$ direction, then the in-context learning classifier*

$$\arg\max_{y \in \mathcal{Y}} P_M^d(X|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y = y)$$

always has a higher or equal probability of misclassification to the Bayes optimal classifier $\arg\max_{y \in \mathcal{Y}} P_M^d(X|\theta^d, Y = y)$. Equality only takes when

$$\forall y \in \mathcal{Y},\ P_M^d(\theta^d|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y = y) = 1.$$

Proof. This theorem can be proved similarly as Theorem A.3. Recall that in Equation (2), we have

$$P_M^d(X|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y) = \int_\Theta P_M^d(X|\boldsymbol{\theta}, Y) P_M^d(\boldsymbol{\theta}|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y) d\boldsymbol{\theta}.$$

By Proposition A.5, $\arg\max_{y \in \mathcal{Y}} P_M^d(X|\theta^d, Y = y)$ is the Bayes optimal classifier. Let $C_{\boldsymbol{\theta}}(X) = \arg\max_{y \in \mathcal{Y}} P_M^d(X|\boldsymbol{\theta}, Y = y)$, then the risk is defined as the probability of misclassification

$$R(C_{\boldsymbol{\theta}}) = P(C_{\boldsymbol{\theta}}(X) \neq Y) = \mathbb{E}_{XY}[\mathbb{1}_{C_{\boldsymbol{\theta}}(X) \neq Y}].$$

Denote the in-context learning classifier $\arg\max_{y \in \mathcal{Y}} P_M^d(X|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y = y)$ by $C_k(X)$. We then have

$$R(C_k) = \mathbb{E}_{XY}[\mathbb{1}_{C_k(X) \neq Y}] = \mathbb{E}_X[\sum_{y \in \mathcal{Y}} (1 - P_M^d(X|\theta^d, Y = y))\mathbb{1}_{C_k(X) = y}].$$

Such risk is minimized if and only if $C_k(X) = C_{\theta^d}(X)$, which only holds when $P_M^d(\theta^d|Y_1^d, X_1^d, ..., Y_k^d, X_k^d, Y = y) = 1$ for all $y \in \mathcal{Y}$. □

A.3. Method

Proposition A.7. *(Proposition 3.1) When $\mathcal{L}(\hat{\theta}^d)$ is minimized, $P_M^d(Y|\hat{\theta}^d, X) = P(Y|\theta^d, X)$ for $X \to Y \leftarrow \boldsymbol{\theta}$, and $P_M^d(X|\hat{\theta}^d, Y) = P(X|\theta^d, Y)$ for $Y \to X \leftarrow \boldsymbol{\theta}$. If the LLM M is invertible, then $\hat{\theta}^d = \theta^d$.*

Proof. The proof of this proposition is straightforward.

Since

$$\mathcal{L}(\hat{\theta}^d) = H(P(Y|\theta^d, X)) + KL(P(Y|\theta^d, X)||P_M^d(Y|\hat{\theta}^d, X))$$

when $\mathcal{L}(\hat{\theta}^d)$ is minimized, we have $P_M^d(Y|\hat{\theta}^d, X) = P(Y|\theta^d, X)$ for $X \to Y \leftarrow \boldsymbol{\theta}$, and $P_M^d(X|\hat{\theta}^d, Y) = P(X|\theta^d, Y)$ for $Y \to X \leftarrow \boldsymbol{\theta}$.

If M is invertible, since the embedding matrix is invertible with or without new concept tokens, $P_M^d(Y|\hat{\theta}, X) = P_M^d(Y|\hat{\theta}', X)$ implies that $\hat{\theta} = \hat{\theta}'$. Thus $\boldsymbol{\theta}$ is identifiable, which means $\hat{\theta}^d = \theta^d$. □

Table 1. Prompt template and label mapping for the datasets we use. Since almost all sentences from ETHOS contain offensive content, we mask out the key offensive words in the examples below.

Dataset	Prompt	Label Mapping
SST-2	sentence: well worth revisiting as many times positive	negative/positive
FPB	The company anticipates its turnover for the whole 2010 to surpass that of the previous year when it was EUR 67.1 million . positive	negative/neutral/positive
COLA	It is this hat that I know the boy who is wearing. unacceptable	acceptable/unacceptable
DBPedia	The Nucet River is a tributary of the Chiojdeanca River in Romania. NaturalPlace	Album/Animal/Artist/ Athlete/Building/Company/ EducationalInstitution/Film/ MeanOfTransportation/ NaturalPlace/OfficeHolder/ Plant/Village/WrittenWork
EmoC	fast i mean fastingis a way of skipping meals i mena you move on too fast others	angry/happy/others/sad
EmoS	i feel this place was tragic sadness	anger/fear/joy/love/ sadness/surprise
ETHOS-SO	[Masked] should be removed from the face of the earth true	false/true
ETHOS-R	I hate being a [Masked], wish I was a [Masked] and no [Masked] on earth existed false	false/true

B. Experiments

Dateset. In Table 1, we show how we process the text classification datasets into prompts. For each dataset, we take at most 16384 examples from the training set for training, and uniformly sample at most 1000 examples from the test set to test the in-context learning performance. In Table 2, we show the train size and test size we used for each dataset. We also list the set of diverse tasks trained with each dataset, which are denoted by their name in Huggingface datasets.[7] The license for SST2, ETHOS-SO and ETHOS-R is GNU General Public License v3. FPB is under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. Note that these two datasets are hate speech detection datasets for different kinds of hate speech and contain many offensive texts. COLA is excerpted from the published works available on the website, and the copyright (where applicable) remains with the original authors or publishers. DBpedia is under a Creative Commons Attribution-ShareAlike License and the GNU Free Documentation License. EmoC and EmoS should be used for educational and research purposes only.

Experiment details. We run our experiments on A100, V100, and A6000 GPUs. We adopt a large portion of the code from the MetaICL repository (Min et al., 2022b)[8]. The training takes around 20 to 40 hours on a single GPU. We use a learning rate of 1e-4 and a batch size of 16, and train for 10k steps in total.

Main results. In Table 3, we list the detailed results of our method and baselines with different LLMs on different datasets in Figure 2.

Causal direction results. The detailed results with anti-causal direction (the opposite direction to what we described in Section 4 are in Table 6) are shown in Table 6, corresponding to Figure 8 in the main text.

Other LLMs results. The detailed results with other LLMs are shown in Table 5, corresponding to Figure 3 in the main text.

Random token results. The detailed results with random tokens are shown in Table 4, corresponding to Figure 4 in the main text.

k-ablation study results. The detailed results of k ablation study are shown in Table 9, corresponding to Figure 5 in the

[7] https://huggingface.co/docs/datasets/index
[8] https://github.com/facebookresearch/MetaICL

datset d	train size	test size	task set $\mathcal{S}$
SST2 (glue-sst2)	16384	1000	glue-cola/glue-mnli/glue-qqp/ glue-mrpc/glue-qnli/glue-rte/glue-sst2/glue-wnli
FPB (financial_phrasebank)	1811	453	glue-sst2/glue-mnli/math_qa/sciq/ social_i_qa/wino_grande/glue-qqp/ ag_news/financial_phrasebank/ poem_sentiment/anli/quarel/quartz/ medical_questions_pairs/paws/dbpedia_14
COLA (cola-sst2)	8551	1000	glue-cola/glue-mnli/glue-qqp/glue-mrpc/ glue-qnli/glue-rte/glue-sst2/glue-wnli
DBpedia (dbpedia_14)	16384	1000	glue-sst2/glue-mnli/math_qa/sciq/ social_i_qa/wino_grande/glue-qqp/ ag_news/financial_phrasebank/ poem_sentiment/anli/quarel/quartz/ medical_questions_pairs/paws/dbpedia_14
EmoC (emo)	16384	1000	glue-sst2/amazon_polarity/ financial_phrasebank/poem_sentiment/ yelp_polarity/glue-cola/blimp/ag_news/ dbpedia_14/ethos/emo/emotion
EmoS (emotion)	16000	1000	glue-sst2/amazon_polarity/ financial_phrasebank/poem_sentiment/ yelp_polarity/glue-cola/blimp/ag_news/ dbpedia_14/ethos/emo/emotion
ETHOS-SO (ethos-sexual_orientation)	346	87	glue-sst2/amazon_polarity/ financial_phrasebank/poem_sentiment/ yelp_polarity/glue-cola/blimp/ag_news/ dbpedia_14/ethos/emo/emotion
ETHOS-R (ethos-religion)	346	87	glue-sst2/amazon_polarity/ financial_phrasebank/poem_sentiment/ yelp_polarity/glue-cola/blimp/ag_news/ dbpedia_14/ethos/emo/emotion

Table 2. Dataset details

main text. In this experiment, we do not reorder the selected demonstrations according to Equation (3), as we need to use GPT2-large for the reordering, and it cannot fit in all the demonstrations. Instead, we order the selected demonstrations from the largest $\hat{P}_M^d(\theta^d|X^d, Y^d)$ to the smallest.

c-ablation study results. The detailed results of c ablation study are shown in Table 10, corresponding to Figure 6 in the main text.

Effect of using ground truth labels. According to (Min et al., 2022c), the ground truth label is not necessary for demonstrations to have a good in-context learning performance, which we found is not entirely true for all the tasks. We compare our method with the randomly selected demonstration baseline under three scenarios: (a) **Original**: demonstrations with the correct labels; (b) **Random words**: using a random label projection map τ^d instead of a meaningful one. i.e., map each label to a fixed random word. In this case, the mapping from the input tokens X to the labels Y is still preserved; (c) **Random labels**: assign a random label to each demonstration, with the original label projection map τ^d. As shown in Figure 9, by using a random label projection map or randomly assigning the labels, the performance of the randomly selected demonstration baseline drops considerably. And randomize the label assignment gives a larger performance drop

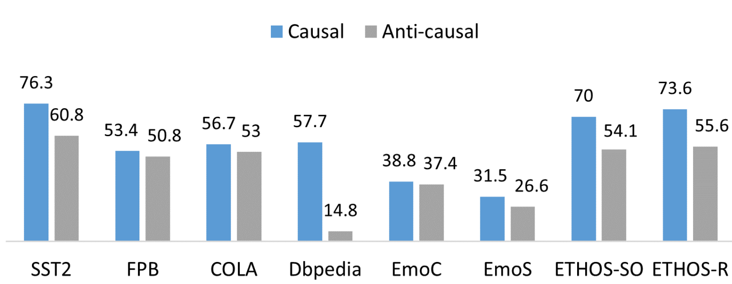

Figure 8. Accuracy of randomly selected demonstrations averaged over seven different LLMs except for GPT3-davinci, using the adopted *causal* direction and the *anti-causal* direction.

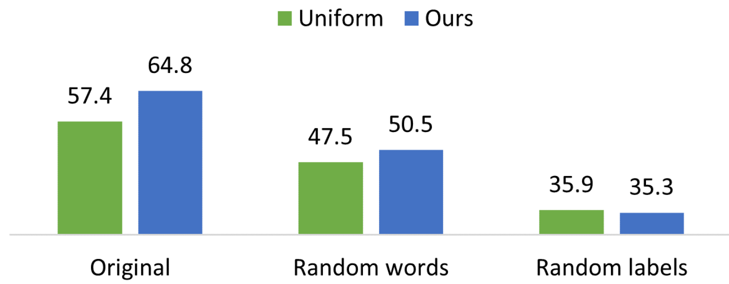

Figure 9. In-context learning accuracy of our method versus random selection baseline, with (a) ground truth labels (*original*), (b) random label mapping (*random words*), or random label assignments (*random label*), averaged over all eight datasets. Numbers are obtained with GPT2-large.

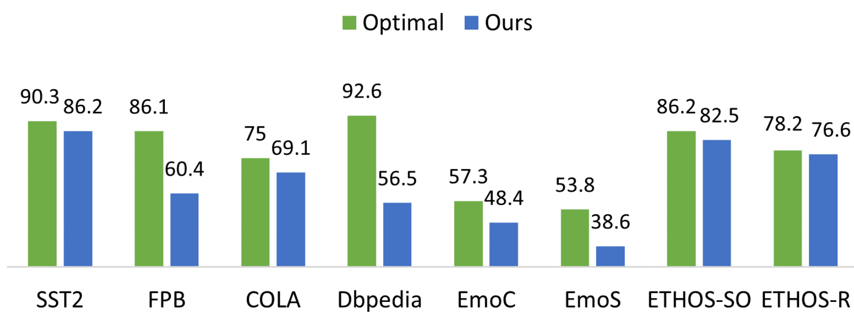

Figure 10. Accuracy of in-context learning using our method versus the theoretical maximum accuracy obtained using the learned concept tokens as prefixes. Numbers are obtained with GPT2-large.

than only using a random label projection map, which shows that the mapping between X and Y in the demonstrations matters. This indicates that in-context learning infers the mapping between X and Y from the demonstrations instead of merely invoking some learned function stored in the LLM parameters based on the appearance of X and Y. We also show that the demonstrations selected by our method represent the $X - Y$ mapping better, as under the **Random words** condition, our method performs better than the random selection baseline, while our method does not improve the random selection baseline under the **Random labels** condition. The detailed results with random words and random labels are shown in Table 7

Optimal performance As stated in Theorem 2.3, the optimal performance of an in-context learning classifier is the Bayes optimal classifier $\arg\max_{y \in \mathcal{Y}} P_M^d(Y = y|\theta^d, X)$, which is approximated by using the learned concept tokens as prefixes. Note that this approximated Bayes optimal classifier cannot be transferred across different LLMs, as the learned concept tokens embeddings are aligned with a specific LLM. The advantage of in-context learning with our method is that the demonstrations can be transferred to any LLMs without training. Here we only compare the accuracy of in-context learning with our method and the approximated Bayes optimal classifier using GPT2-large, as it is the LLM that concept tokens are fine-tuned with. As shown in Figure 10, our method comes close to the optimal accuracy on many datasets, while there are some datasets that our method is lagging. This indicates that there are two ways to improve our method: the first is to improve the performance of the optimal classifier, by introducing a better latent concept learning algorithm. The other way is to reduce the performance gap between our method and the optimal classifier, by improving the demonstration selection algorithm. The detailed results using the learned concept tokens as prefixes are shown in Table 8.

Reordering results. The detailed results with and without reordering are shown in Table 11, corresponding to Figure 11.

Similar tokens. We show the top ten similar tokens to some learned concept tokens in Table 12, as summarized in Figure 7 in the main text.

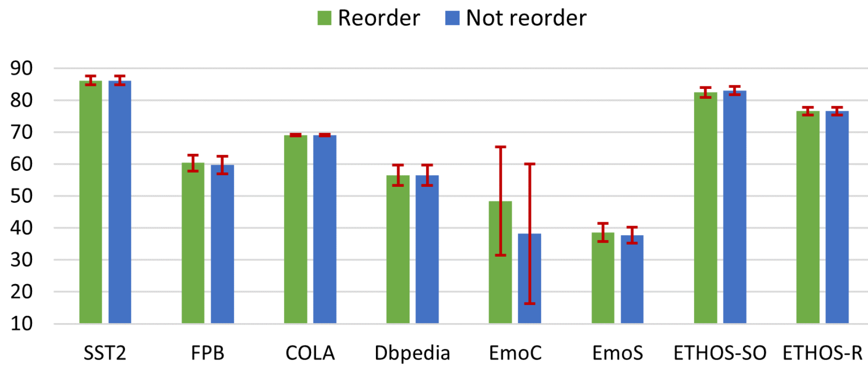

Figure 11. In-context learning accuracy of our method versus random selection baseline, with and without reordering. The red error bars represent the standard deviation across five runs. Numbers are obtained with GPT2-large.

C. Limitations and Future Work

While the assumption that a large language model captures the true distribution of language is fairly common in the literature studying LLMs (Xie et al., 2022; Saunshi et al., 2021), this assumption is not entirely accurate in practice. According to (LeBrun et al., 2022), LLMs systematically underestimate rare text sequences, which constitute a significant portion of the long-tail distribution of language. Although this assumption is adequate to achieve favorable empirical results, it is expected that more accurate language models will, in theory, lead to improved outcomes.

The selection of the accompanying diverse tasks $\mathcal{S}$ is currently left to the user's discretion. A better approach to constructing such a task set is needed to gain a deeper understanding of latent concept variables and to improve the latent concept learning algorithm.

Our algorithm currently only applies to classification tasks. More complex latent variables could be designed to improve the in-context learning performance of more complex tasks like math word questions and logical reasoning problems.

D. Broader Impact

The utilization of language models (LLMs) for specific tasks is often hindered by the high cost associated with training or fine-tuning them. However, the in-context learning paradigm offers a cost-effective and convenient alternative for utilizing the power of pre-trained LLMs. Our work has demonstrated a significant improvement in the performance of in-context learning through a relatively low-cost and simple approach, thus making the use of LLMs more accessible for individuals with limited resources.

However, it is important to consider the broader implications of the increasing use of LLMs. As LLMs are not infallible and may make mistakes, it is crucial to explicitly warn users of the potential for misleading output and to regulate the distribution of LLMs in order to prevent any negative societal impact. Additionally, it is possible that LLMs could be intentionally misused, thus it is important to consider the ethical implications of their use and to take appropriate measures to mitigate any potential negative effects. We posit that these regulations and measures should be put in place at the time of distributing LLMs to ensure the safe and responsible use of these models. Furthermore, as we publicly release our code, we will also provide clear warnings and guidelines to users to ensure that the potential risks associated with the use of our method are fully understood and addressed.

Table 3. Accuracy of selected demonstration. Our demonstrations are selected using GPT2-large, and the same set of demonstrations is applied to all different LLMs. All LLMs are pre-trained only with the language modeling objective, while the pre-training data size of GPT2s is much smaller than GPT3s.

LLM	Method	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
GPT2	Uniform	69.7 ± 1.8	52.9 ± 2.3	61.9 ± 1.4	48.0 ± 0.7	35.3 ± 1.7	26.4 ± 1.0	64.1 ± 4.8	71.0 ± 1.8	53.7
(124M)	Similar	69.5 ± 0.6	55.9 ± 1.7	63.2 ± 1.2	44.7 ± 3.1	36.4 ± 2.0	26.6 ± 1.3	77.7 ± 2.7	80.0 ± 3.7	56.8
	Ours	76.8 ± 2.9	64.5 ± 3.2	69.1 ± 0.2	53.5 ± 2.95	37.2 ± 11.1	30.6 ± 4.8	80.9 ± 1.9	76.8 ± 2.6	61.2
GPT2-m	Uniform	70.8 ± 1.3	52.0 ± 1.7	57.8 ± 1.3	49.3 ± 2.0	34.2 ± 1.8	34.2 ± 1.8	76.3 ± 4.9	74.7 ± 2.2	56.2
(355M)	Similar	75.0 ± 1.9	57.7 ± 2.0	57.5 ± 2.2	47.9 ± 6.0	37.2 ± 3.6	35.2 ± 1.8	86.9 ± 2.9	84.6 ± 4.3	60.3
	Ours	81.2 ± 1.3	59.3 ± 4.3	69.0 ± 0.2	52.9 ± 2.3	40.4 ± 21.5	37.2 ± 2.4	83.7 ± 1.1	76.8 ± 1.1	62.6
GPT2-l	Uniform	77.1 ± 1.2	51.3 ± 2.4	62.7 ± 0.8	54.4 ± 0.9	38.7 ± 2.1	34.5 ± 1.2	67.6 ± 4.3	72.9 ± 2.8	57.4
(774M)	Similar	80.7 ± 1.6	54.8 ± 3.8	50.9 ± 1.4	51.1 ± 5.2	39.9 ± 2.6	35.1 ± 2.1	80.9 ± 2.8	84.4 ± 2.6	59.7
	Ours	86.2 ± 1.4	60.4 ± 2.5	69.1 ± 0.2	56.5 ± 3.2	48.4 ± 17.0	38.6 ± 2.8	82.5 ± 1.5	76.6 ± 1.2	64.8
GPT2-xl	Uniform	74.7 ± 0.9	53.2 ± 1.9	55.8 ± 1.6	53.0 ± 1.9	38.2 ± 1.5	38.2 ± 1.5	67.8 ± 6.4	72.6 ± 4.1	56.7
(1.5B)	Similar	80.6 ± 1.3	53.0 ± 2.5	55.0 ± 2.5	51.6 ± 5.9	39.9 ± 2.0	32.9 ± 2.1	82.8 ± 2.2	83.9 ± 4.5	60
	Ours	83.1 ± 3.6	62.0 ± 2.5	68.9 ± 0.2	58.6 ± 3.3	43.6 ± 16.4	43.6 ± 16.4	83.0 ± 1.3	77.9 ± 1.3	65.1
GPT3-a	Uniform	76.9 ± 0.9	56.6 ± 1.1	53.1 ± 1.8	62.1 ± 1.4	38.6 ± 1.4	27.7 ± 1.3	65.5 ± 5.7	74.0 ± 3.0	56.8
(350M)	Similar	78.7 ± 1.0	52.2 ± 2.7	53.1 ± 1.8	54.6 ± 1.7	42.4 ± 3.5	37.2 ± 1.1	84.1 ± 2.2	87.8 ± 3.5	61.3
	Ours	85.4 ± 1.7	61.9 ± 10.5	58.2 ± 7.0	64.0 ± 4.4	43.0 ± 7.2	37.9 ± 2.3	84.4 ± 1.4	78.9 ± 0.9	64.2
GPT3-b	Uniform	80.8 ± 0.6	55.2 ± 3.3	46.8 ± 2.0	66.5 ± 1.4	42.0 ± 0.7	27.0 ± 1.2	71.0 ± 4.6	72.6 ± 3.1	57.7
(1.3B)	Similar	83.9 ± 1.3	56.2 ± 2.3	45.1 ± 1.8	59.8 ± 1.8	42.9 ± 3.5	38.1 ± 1.7	86.7 ± 3.0	86.4 ± 3.0	62.4
	Ours	87.3 ± 2.0	64.3 ± 5.9	67.2 ± 0.9	70.2 ± 3.2	43.6 ± 13.0	38.9 ± 5.0	84.6 ± 0.9	78.9 ± 1.2	66.9
GPT3-c	Uniform	84.2 ± 1.4	52.6 ± 1.8	59.1 ± 1.5	70.6 ± 0.8	44.3 ± 2.5	32.3 ± 1.9	77.5 ± 4.7	77.5 ± 0.6	62.3
(6.7B)	Similar	85.7 ± 1.4	62.2 ± 0.9	58.0 ± 1.7	62.2 ± 2.0	47.4 ± 4.3	39.8 ± 1.7	89.2 ± 1.4	89.7 ± 1.9	66.8
	Ours	88.8 ± 0.7	64.1 ± 5.7	69.0 ± 0.3	73.6 ± 2.9	50.3 ± 11.9	43.1 ± 4.6	86.2 ± 0.0	78.2 ± 0.0	69.2
GPT3-d	Uniform	86.5 ± 0.9	59.2 ± 2.4	45.5 ± 2.8	73.6 ± 1.9	39.4 ± 0.7	40.6 ± 1.7	77.2 ± 2.6	76.8 ± 3.5	62.4
(175B)	Similar	88.5 ± 0.8	55.4 ± 3.3	45.4 ± 1.5	67.2 ± 1.8	37.6 ± 1.6	39.8 ± 1.4	86.9 ± 2.4	89.0 ± 3.8	63.7
	Ours	87.8 ± 3.4	62.7 ± 3.3	58.5 ± 8.2	75.5 ± 2.4	41.3 ± 3.6	42.7 ± 3.9	85.1 ± 0.0	79.3 ± 0.0	66.6
Avg	Uniform	77.6	54.1	55.3	59.7	38.8	32.6	70.9	74.0	57.9
	Similar	80.3	55.9	53.5	54.9	40.5	35.6	84.4	85.7	61.4
	Ours	84.6	62.4	66.1	63.1	43.5	39.1	83.8	77.9	65.0

Table 4. Accuracy of selected demonstration. Our demonstrations are selected using GPT2-large, and the same set of demonstrations is applied to all different LLMs. All LLMs are pre-trained only with the language modeling objective, while the pre-training data size of GPT2s is much smaller than GPT3s.

LLM	Method	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
GPT2	Uniform	69.7 ± 1.8	52.9 ± 2.3	61.9 ± 1.4	48.0 ± 0.7	35.3 ± 1.7	26.4 ± 1.0	64.1 ± 4.8	71.0 ± 1.8	53.7
(124M)	Random	69.8 ± 3.3	51.1 ± 1.7	69.0 ± 0.1	49.0 ± 4.5	33.7 ± 15.5	24.2 ± 7.6	66.4 ± 17.5	66.2 ± 16.2	53.7
	Ours	76.8 ± 2.9	64.5 ± 3.2	69.1 ± 0.2	53.5 ± 2.95	37.2 ± 11.1	30.6 ± 4.8	80.9 ± 1.9	76.8 ± 2.6	61.2
GPT2-l	Uniform	77.1 ± 1.2	51.3 ± 2.4	62.7 ± 0.8	54.4 ± 0.9	38.7 ± 2.1	34.5 ± 1.2	67.6 ± 4.3	72.9 ± 2.8	57.4
(774M)	Random	81.9 ± 4.5	46.5 ± 4.7	64.9 ± 7.8	50.3 ± 4.3	42.5 ± 16.7	36.1 ± 6.5	67.6 ± 20.4	67.8 ± 15.0	57.2
	Ours	86.2 ± 1.4	60.4 ± 2.5	69.1 ± 0.2	56.5 ± 3.2	48.4 ± 17.0	38.6 ± 2.8	82.5 ± 1.5	76.6 ± 1.2	64.8

Table 5. We test our method on other similar sizes (6-7B) LLMs.

LLM	Method	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
GPT2-l	Random	77.1 ± 1.2	51.3 ± 2.4	62.7 ± 0.8	54.4 ± 0.9	38.7 ± 2.1	34.5 ± 1.2	67.6 ± 4.3	72.9 ± 2.8	57.4
	Ours	86.2 ± 1.4	60.4 ± 2.5	69.1 ± 0.2	56.5 ± 3.2	48.4 ± 17.0	38.6 ± 2.8	82.5 ± 1.5	76.6 ± 1.2	64.8
GPT3-c	Random	84.2 ± 1.4	52.6 ± 1.8	59.1 ± 1.5	70.6 ± 0.8	44.3 ± 2.5	32.3 ± 1.9	77.5 ± 4.7	77.5 ± 0.6	62.3
	Ours	88.8 ± 0.7	64.1 ± 5.7	69.0 ± 0.3	73.6 ± 2.9	50.3 ± 11.9	43.1 ± 4.6	86.2 ± 0.0	78.2 ± 0.0	69.2
GPT-J	Random	78.5 ± 1.0	53.1 ± 1.7	58.3 ± 2.2	55.6 ± 1.2	38.5 ± 2.0	33.3 ± 1.5	76.6 ± 3.7	76.6 ± 1.4	58.8
	Ours	87.8 ± 1.9	56.7 ± 4.3	69.1 ± 0.2	60.0 ± 3.6	32.5 ± 16.1	33.2 ± 2.8	85.3 ± 0.5	77.0 ± 0.0	62.7
OPT	Random	72.4 ± 0.8	32.8 ± 0.3	34.8 ± 0.6	29.4 ± 1.4	67.1 ± 1.8	36.9 ± 0.6	86.2 ± 0.0	78.2 ± 0.0	54.7
	Ours	74.2 ± 3.0	34.1 ± 6.1	35.7 ± 3.1	28.8 ± 2.1	76.7 ± 4.1	39.0 ± 3.4	86.2 ± 0.0	78.2 ± 0.0	56.6
LLaMA	Random	57.7 ± 1.5	23.7 ± 1.3	30.8 ± 0.2	15.8 ± 0.8	4.4 ± 0.7	35.2 ± 0.7	66.2 ± 5.8	57.2 ± 5.1	36.4
	Ours	60.5 ± 4.7	19.1 ± 1.9	30.8 ± 0.2	16.9 ± 1.3	4.3 ± 0.7	35.3 ± 0.6	77.2 ± 13.6	56.3 ± 10.8	37.6

Table 6. We test random selection baseline with anti-causal direction.

LLM	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R
GPT2	57.4 ± 1.9	56.6 ± 2.1	55.9 ± 1.7	11.3 ± 1.0	24.6 ± 2.4	22.1 ± 1.1	64.1 ± 4.8	58.6 ± 5.5
GPT2-m	56.7 ± 1.6	48.7 ± 2.1	55.3 ± 1.8	13.9 ± 1.2	22.4 ± 1.9	24.9 ± 2.3	44.8 ± 1.9	45.5 ± 3.5
GPT2-l	58.7 ± 0.7	33.7 ± 1.3	50.8 ± 1.6	13.6 ± 1.3	28.2 ± 3.6	26.2 ± 2.7	48.7 ± 3.7	53.6 ± 5.3
GPT2-xl	54.2 ± 0.5	46.8 ± 1.2	50.6 ± 1.1	12.6 ± 1.5	31.4 ± 2.8	25.9 ± 3.2	65.5 ± 4.9	61.8 ± 1.5
GPT3-a	55.8 ± 0.9	58.9 ± 2.1	51.6 ± 1.4	14.3 ± 0.8	54.2 ± 3.1	27.7 ± 1.3	49.2 ± 3.3	54.9 ± 6.4
GPT3-b	64.4 ± 1.6	58.9 ± 2.6	53.4 ± 1.1	14.6 ± 1.1	52.0 ± 2.5	27.0 ± 1.3	48.3 ± 2.7	51.0 ± 4.0
GPT3-c	78.2 ± 1.6	52.3 ± 2.3	53.7 ± 0.7	23.0 ± 2.5	49.1 ± 2.6	32.2 ± 1.9	57.9 ± 2.7	64.1 ± 5.0
Avg	60.8	50.8	53	14.8	37.4	26.6	54.1	55.6

Table 7. We test our method with random words and random labels using GPT2-large.

	Method	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
R words	Random	54.1 $\pm$ 4.2	43.4 $\pm$ 1.9	62.2 $\pm$ 4.9	11.2 $\pm$ 0.9	32.4 $\pm$ 5.2	19.1 $\pm$ 1.8	80.7 $\pm$ 4.8	77.0 $\pm$ 3.6	47.5
	Ours	50.3 $\pm$ 1.3	44.9 $\pm$ 4.2	69.2 $\pm$ 0.2	13.9$\pm$1.2	37.8 $\pm$ 12.1	23.5 $\pm$ 7.4	86.0 $\pm$ 0.5	77.9 $\pm$ 0.5	50.5
R labels	Random	51.5 $\pm$ 0.9	32.5 $\pm$ 1.2	49.3 $\pm$ 3.0	6.7 $\pm$ 1.0	25.1 $\pm$ 0.6	17.2 $\pm$ 0.9	48.0 $\pm$ 2.5	56.8 $\pm$ 3.1	35.9
	Ours	49.6 $\pm$ 0.9	36.2 $\pm$ 2.5	49.3 $\pm$ 1.6	6.6$\pm$ 0.2	24.7 $\pm$ 0.6	16.6 $\pm$ 1.0	51.0 $\pm$ 4.9	48.7 $\pm$ 3.5	35.3

Table 8. Accuracy using concept tokens as prefixes.

SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R
90.3 $\pm$ 0.0	86.1 $\pm$ 0.0	75.0 $\pm$ 0.1	92.6 $\pm$ 0.6	57.3 $\pm$ 1.8	53.8 $\pm$ 0.7	86.2 $\pm$ 0.0	78.2 $\pm$ 0.0

Table 9. k ablation study using GPT2-large, without reordering.

	Method	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
$k = 2$	Random	74.4 $\pm$ 1.0	48.5 $\pm$ 1.1	48.9 $\pm$ 1.6	52.9 $\pm$ 2.0	42.8 $\pm$ 0.6	37.1 $\pm$ 1.2	66.9 $\pm$ 4.7	66.4 $\pm$ 6.8	54.7
	Ours	78.1 $\pm$ 4.5	50.1 $\pm$ 2.9	54.3 $\pm$ 8.8	57.3 $\pm$ 5.1	41.1 $\pm$ 9.8	36.1 $\pm$ 2.6	84.6 $\pm$ 1.6	76.8 $\pm$ 4.5	59.8
$k = 4$	Random	76.9 $\pm$ 0.7	56.6 $\pm$ 1.1	53.1 $\pm$ 1.8	62.1 $\pm$ 1.4	38.6 $\pm$ 1.4	27.7 $\pm$ 1.3	65.5 $\pm$ 5.7	74.0 $\pm$ 3.0	56.8
	Ours	86.2 $\pm$ 1.4	59.7 $\pm$ 2.8	69.1 $\pm$ 0.2	56.5 $\pm$ 3.2	38.2 $\pm$ 21.8	37.7 $\pm$ 2.5	83.0 $\pm$ 1.3	76.6 $\pm$ 1.2	63.4
$k = 8$	Random	79.9 $\pm$ 0.2	57.1 $\pm$ 1.6	51.3 $\pm$ 1.0	66.5 $\pm$ 1.2	37.6 $\pm$ 1.5	36.2 $\pm$ 0.6	68.5 $\pm$ 3.5	72.9 $\pm$ 3.3	58.8
	Ours	87.0 $\pm$ 2.4	59.9 $\pm$ 3.3	55.3 $\pm$ 9.7	67.0 $\pm$ 0.9	39.9 $\pm$ 5.3	38.8 $\pm$ 2.6	77.0 $\pm$ 11.1	78.9 $\pm$ 0.9	63
$k = 16$	Random	79.9 $\pm$ 1.1	54.9 $\pm$ 2.7	54.5 $\pm$ 2.8	69.1 $\pm$ 1.1	33.7 $\pm$ 2.2	33.5 $\pm$ 1.4	64.8 $\pm$ 4.0	69.0 $\pm$ 3.2	57.4
	Ours	84.6 $\pm$ 1.9	60.4 $\pm$ 6.4	62.0 $\pm$ 7.0	71.0 $\pm$ 1.9	37.2 $\pm$ 6.1	37.1 $\pm$ 2.2	72.4 $\pm$ 7.6	74.7 $\pm$ 4.7	62.4

Table 10. c ablation study using GPT2-large

	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
$c = 5$	78.9 $\pm$ 2.4	59.8 $\pm$ 10.8	34.3 $\pm$ 5.0	62.9 $\pm$ 2.4	44.9 $\pm$ 9.5	38.1 $\pm$ 2.4	71.7 $\pm$ 5.9	62.1 $\pm$ 19.7	56.6
$c = 10$	85.4 $\pm$ 1.7	61.9 $\pm$ 10.5	58.2 $\pm$ 7.0	64.0 $\pm$ 4.4	43.0 $\pm$ 7.2	37.9 $\pm$ 2.3	84.4 $\pm$ 1.4	78.9 $\pm$ 0.9	64.2
$c = 15$	80.1 $\pm$ 1.4	64.3 $\pm$ 7.7	63.1 $\pm$ 9.4	58.7 $\pm$ 3.2	36.4 $\pm$ 11.5	38.6 $\pm$ 1.9	80.9 $\pm$ 3.9	76.3 $\pm$ 5.9	62.3
$c = 20$	78.5 $\pm$ 4.1	51.8 $\pm$ 8.0	66.5 $\pm$ 2.3	58.0 $\pm$ 3.4	36.3 $\pm$ 4.3	41.8 $\pm$ 5.8	80.7 $\pm$ 4.5	73.8 $\pm$ 5.4	60.92

Table 11. Reorder versus not reorder using our method, with GPT2-large.

	SST2	FPB	COLA	DBpedia	EmoC	EmoS	ETHOS-SO	ETHOS-R	Avg
reorder	86.2 $\pm$ 1.4	60.4 $\pm$ 2.5	69.1 $\pm$ 0.2	56.5 $\pm$ 3.2	48.4 $\pm$ 17.0	38.6 $\pm$ 2.8	82.5 $\pm$ 1.5	76.6 $\pm$ 1.2	64.8
not reorder	86.2 $\pm$ 1.4	59.7 $\pm$ 2.8	69.1 $\pm$ 0.2	56.5 $\pm$ 3.2	38.2 $\pm$ 21.8	37.7 $\pm$ 2.5	83.0 $\pm$ 1.3	76.6 $\pm$ 1.2	63.4

Table 12. We list the top 10 similar words (tokens) to some of the learned concept tokens.

concept token	similar words
FPB-2	milo coordinate notify rendering benefiting routing EntityItem routed Messages Plot
FPB-3	unlocked updating deleting dropping damage updates drops Gained taken dropped
FPB-4	FX Safari Fixes advertisers Links Coins Operator marketers Guidelines
FPB-5	674 592 693 696 498 593 793 504 691 683
COLA-1	exha trunc curv fragmented elong iterator initialized bounds Iter filament
COLA-2	Sp spa contributed cerv borrower paper tiger Erica USH Schwartz
COLA-7	democr Barack WH ophobic neum Democrats Rachel WH Democrats
DBpedia-4	often impede blockade incarcerated LEASE pollutants pesticides uphe lawmakers fossils
DBpedia-5	categorized closes therapies antidepressant retrospective clinically physicians therapists randomized clinicians
DBpedia-7	JS provided Killed richness Compet Nevertheless Probably Proceedings horizontally
ETHOS-SO-3	Revolution Spread itu Million Pascal stabil Indy Georgian Figure resy
ETHOS-R-2	council Chocobo Shant uyomi aditional cumbers subur ThumbnailImage araoh Pharaoh
ETHOS-R-8	seems outlines emitted grin outline circuitry sized flips emits flipped
ETHOS-R-9	223 asel Cyrus Sith Scorpion Snape Jas Leia Ned Morty
EmoC-6	behavi checkpoints unintention crib eleph looph np mosquit blat pione
EmoC-8	depressed bullied choked stricken devastated unsuccessful cheated distraught troubled failing
EmoS-1	frightened rebellious depressed careless bullied restless reluctant distraught clumsy disgruntled
EmoS-5	obsessive crappy demonic delusions psychosis psychotic childish stupidity reckless insanity
EmoS-7	benevolent charismatic perfected volunte unintention pione innocuous fearless glamorous ruthless
EmoS-9	whispers pundits Sadly horribly curiously noticeably Sadly gaping painfully shockingly