Knowledgeable In-Context Tuning: Exploring and Exploiting Factual Knowledge for In-Context Learning

Jianing Wang[1]*, **Chengyu Wang**[2]*, **Chuanqi Tan**[2], **Jun Huang**[2], **Ming Gao**[1,3]†
[1] School of Data Science and Engineering, East China Normal University, Shanghai, China
[2] Alibaba Group, Hangzhou, China
[3] KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai, China
lygwjn@gmail.com, {chengyu.wcy, chuanqi.tcq}@alibaba-inc.com
huangjun.hj@alibaba-inc.com, mgao@dase.ecnu.edu.cn

Abstract

Large language models (LLMs) enable in-context learning (ICL) by conditioning on a few labeled training examples as a text-based prompt, eliminating the need for parameter updates and achieving competitive performance. In this paper, we demonstrate that *factual knowledge* is imperative for the performance of ICL in three core facets: the inherent knowledge learned in LLMs, the factual knowledge derived from the selected in-context examples, and the knowledge biases in LLMs for output generation. To unleash the power of LLMs in few-shot learning scenarios, we introduce a novel **K**nowledgeable **I**n-**C**ontext **T**uning (**KICT**) framework to further improve the performance of ICL: 1) injecting knowledge into LLMs during continual self-supervised pre-training, 2) judiciously selecting the examples for ICL with high knowledge relevance, and 3) calibrating the prediction results based on prior knowledge. We evaluate the proposed approaches on autoregressive models (e.g., GPT-style LLMs) over multiple text classification and question-answering tasks. Experimental results demonstrate that **KICT** substantially outperforms strong baselines and improves by more than 13% and 7% on text classification and question-answering tasks, respectively [1].

1 Introduction

Large language models (LLMs) have become an imperative infrastructure in the natural language processing (NLP) community (Zhao et al., 2023b). To enable pre-trained LLMs to perform well without any parameter updates, in-context learning (ICL) has emerged as one of the flourishing research topics in many few-shot NLP tasks. It aims to generate predictions for target examples by conditioning on a few labeled samples (Brown et al., 2020). As shown in Figure 1, the key component of ICL is the text-based prompt (containing labeled examples) that functions as the demonstration.

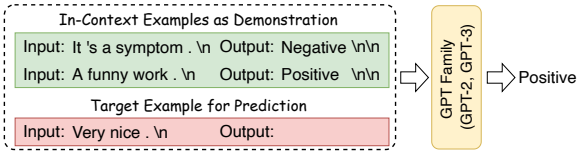


Figure 1: An example of in-context learning (ICL).

Previous works have explored multiple aspects that affect the performance of ICL (Dong et al., 2023), such as input-output mapping (Min et al., 2022b; Kim et al., 2022), extensive data resources (Mishra et al., 2022; Chen et al., 2022b; Min et al., 2022a), prediction calibration (Zhao et al., 2021), and self-improvment (Chen et al., 2023; Lyu et al., 2023). Liu et al. (2022); Lu et al. (2022) have investigated others, such as prompt format (e.g., "Input:", "Output:"), the selection of labeled data, and example permutation. Wang et al. (2023a); Wu et al. (2023) have developed toolkits for LLMs to reason with ICL prompts. In addition, to better elicit the LLM to reason on complex tasks, chain-of-thought (CoT) has been introduced to extend the ICL with multiple rationales to express the thinking process (Wei et al., 2022; Dhuliawala et al., 2023; Wang et al., 2023c,b; Zhao et al., 2023a; Zhang et al., 2023; Liang et al., 2023). However, these works pay little attention to the influence of *factual knowledge* in ICL, which is a non-negligible factor in NLP (Hu et al., 2022).

To this end, we explore the effectiveness of ICL from the perspective of *factual knowledge*. As seen in Figure 2, when entities and labels in text-based prompts are randomly replaced or removed, the average accuracy decreases significantly, indicating that performance degradation is universal across different model scales. Further analysis reveals that: 1) more intrinsic factual knowledge acquired during the pre-training stage is typically beneficial

* J. Wang and C. Wang contributed equally to this work.
† Corresponding author.
[1]The code and datasets are released in HugNLP (Wang et al., 2023a): https://github.com/HugAILab/HugNLP.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3261–3280
June 16-21, 2024

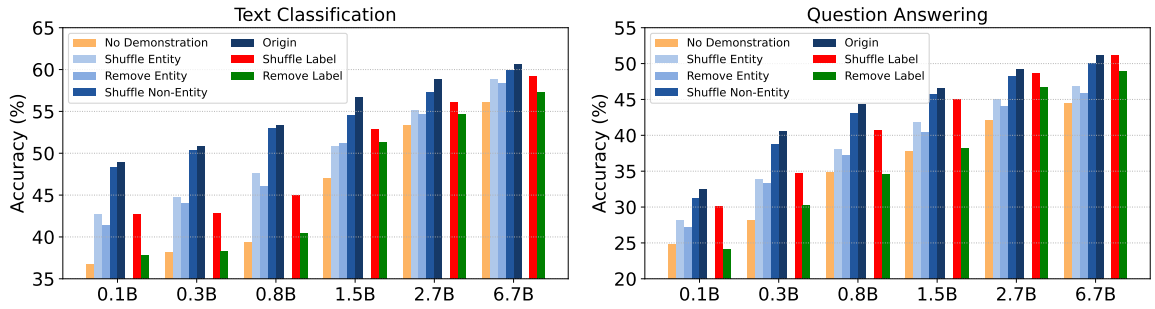


Figure 2: Results of different scales of GPT-2 and OPT models over 8 text classification tasks and 4 question answering tasks in various component destruction settings. For each target example, we have $K = 8$ labeled samples as the demonstration. Results indicate that factual knowledge is crucial to the performance of ICL.

for LLMs to improve effectiveness; 2) The factual knowledge (e.g., entities and labels) derived from selected in-context examples is crucial for the performance of ICL; 3) LLMs tend to generate common words that may have high frequencies in the training corpora, resulting in biased predictions.

After analyzing these knowledge facets, a natural question arises: *How can we fully employ factual knowledge to further improve the performance of ICL?* To achieve this goal, we focus on causal autoregressive LLMs (e.g., GPT-2 (Radford et al., 2019) and OPT (Zhang et al., 2022a)) and present a novel **K**nowledgeable **I**n-**C**ontext **T**uning (**KICT**) framework, which involves knowledgeable guidance in *pre-training*, *prompting*, and *prediction* of these models. Specifically, to endow LLMs with enhanced text generation abilities by better leveraging inherent knowledge, we introduce several knowledgeable self-supervised tasks during the *pre-training* stage to inject knowledge into LLMs. For text-based *prompting*, we propose a knowledgeable example retrieval algorithm to judiciously select in-context examples that have relevant knowledge to the target example. Finally, during *prediction*, we utilize the knowledge-wise priors of label words from an underlying knowledge base (KB) to calibrate the prediction distributions generated by LLMs. Each of the proposed techniques is plug-and-play and can be freely combined, facilitating users to exploit knowledge for improving ICL.

To evaluate the effectiveness of the **KICT** framework, we employ LLMs (e.g., GPT-style models) to conduct extensive experiments over multiple text classification and question-answering tasks. Results demonstrate that each proposed procedure achieves substantial improvements.

To sum up, we make the following main contributions:

- We study three knowledge facets for ICL that are imperative for LLMs in few-shot learning, i.e., inherent knowledge in LLMs, relevant knowledge in the text-based prompt, and knowledge bias.
- We present a novel knowledgeable in-context tuning framework for better incorporating knowledge through the process of pre-training, prompting, and predicting.
- Extensive experiment results show that our approach attains more impressive performance over classification and QA tasks.

2 Impact of Knowledge on ICL

In this section, we investigate whether *factual knowledge* affects the performance of ICL.

2.1 Preliminary Experimental Settings

Following Min et al. (2022b) and Kim et al. (2022), we perform empirical experiments through component destruction. Specifically, given a target example text X^{tgt}, we randomly select K training samples $\widetilde{\mathcal{D}} = \{(X_i^{trn}, y_i^{trn})\}_{i=1}^{K}$ to form a text-based prompt. We identify all entities in the prompt and then devise several destruction settings as follows: 1) `Shuffle Entity` involves randomly replacing all entities with others from the KB; 2) `Shuffle Non-Entity` entails replacing some non-entity words (e.g., "It", "have") with others from the vocabulary; 3) `Shuffle Label` consists of replacing all the golden labels with incorrect ones; 4) `Remove Entity` and `Remove Label` aim to remove all entities and labels from the prompt, respectively; 5) `No Demonstration` represents a typical zero-shot

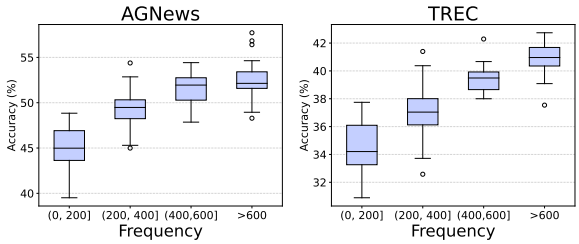


Figure 3: 4-shot results of GPT-2 (urge) over AGNews and TREC. For each frequency region, we sample top-5 label words for each category and report the accuracy for all label mapping permutations.

method where no labeled data is used (Min et al., 2022b).

We employ various scales of GPT-2 (0.1B-1.5B) and OPT (Zhang et al., 2022a) (2.7B-6.7B) models to evaluate 8 text classification tasks and 4 question answering tasks. [2] By default, we randomly sample $K = 8$ labeled samples for each task and conduct the experiments with 5 different random seeds. Further details are presented in Appendix A. The findings are summarized below.

2.2 Findings

The inherent knowledge in the LLM itself is beneficial for the performance of downstream tasks. As shown in Figure 2, models can achieve remarkable few-shot performance with increased scale. We hypothesize that larger models can learn more valuable semantics in the pre-training corpus, which contributes to this improvement. To test this hypothesis, we perform zero-shot inference without any text-based prompts (i.e., `No Demonstration`), relying solely on the intrinsic knowledge acquired during pre-training to guide the predictions. We observe that the performance gap between the 6.7B and 0.1B models is about 20% on both text classification and question-answering tasks. This observation supports the idea that the inherent knowledge learned during pre-training is critical (Yang et al., 2021).

The factual knowledge in selected in-context examples is crucial for ICL. As shown in Figure 2, the original setting (`Origin`) outperforms other configurations across all model scales. We observe that altering non-entity words does not significantly reduce performance, whereas replacing or removing entities leads to a considerable decrease in average accuracy for both text classification and question-answering tasks. This demonstrates that factual knowledge embedded in text-based prompts is a critical factor for LLMs to understand the task. Furthermore, we find that labels are also essential for ICL, echoing similar observations presented in (Kim et al., 2022). Differing from Min et al. (2022b), we posit that labels can be regarded as a form of factual knowledge that guides the LLM to grasp semantics during inference.

LLMs tend to generate common label words due to knowledge bias. To investigate whether predictions are biased, we select two knowledge-intensive tasks (i.e., AGNews (Zhang et al., 2015), and TREC (Voorhees and Tice, 2000)). We first retrieve the top-5 predictions at the output position for each training example[3] and compute frequency statistics for each generated label word. Subsequently, we select 4 labeled examples from the training set for each category. From each frequency region, we randomly choose 2 label words and calculate the average accuracy across all label mapping permutations.[4] The results, as presented in Figure 3, reveal that performance is highly contingent on label word frequency, suggesting that the frequency with which factual knowledge is learned by LLMs plays a critical role in prediction outcomes. Similar observations have been reported by Zhao et al. (2021).

3 The Proposed KICT Framework

The preliminary experiments demonstrate that *factual knowledge* has a substantial effect on ICL. This suggests that we can exploit this knowledge to enhance performance across various processes in ICL, including *pre-training*, *prompting*, and *prediction*. To achieve this goal, we introduce the **KICT** framework, a novel **K**nowledgeable **I**n-**C**ontext **T**uning framework designed to better leverage knowledge and unleash the power of LLMs in answer generation. Within this framework, we introduce Knowledgeable Pre-Training (KPT) with three carefully designed self-supervised tasks to infuse LLMs with factual knowledge. We then present a Knowledgeable Example Retrieval (KER) algorithm to judiciously select in-context examples that are relevant to the given knowledge. Finally, we employ a

[2] Due to resource constraints, we do not use larger models. Nevertheless, our findings are generally consistent across different model scales.

[3] The training set is larger than the testing set, thereby providing a more robust statistical representation.

[4] Considering AGNews as an example, which has 4 classes with 2 label words each, there are $2^4 = 16$ possible label mapping permutations.

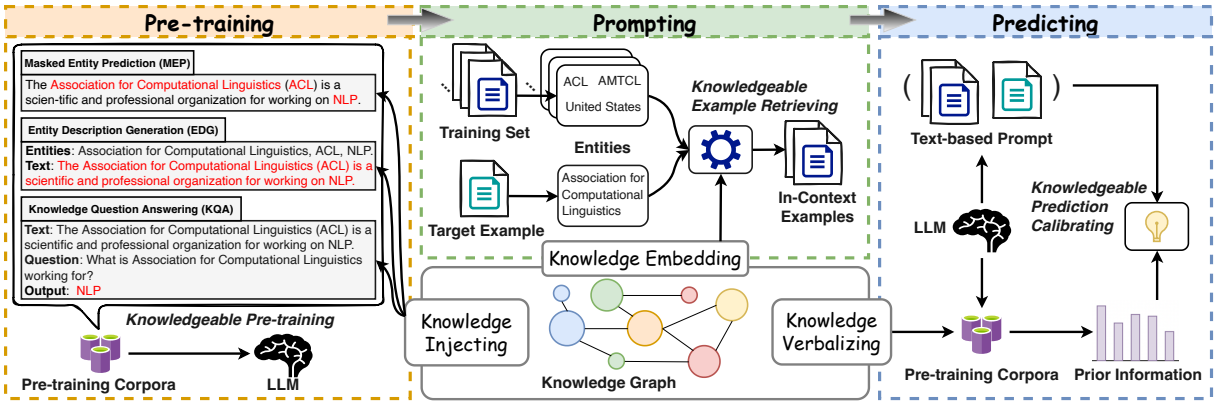


Figure 4: The overview of the **KICT** framework. We introduce multiple plug-and-play knowledgeable techniques to enhance the utilization of knowledge for improving ICL performance. **Left**: We propose three knowledge-aware self-supervised learning tasks that infuse factual knowledge into LLMs during pre-training. **Middle**: We utilize entity-related information to select in-context examples that exhibit high knowledge relevance to the target example. **Right**: For prediction, we derive prior information from large-scale corpora to calibrate the predictions.

Knowledgeable Prediction Calibration (KPC) technique to adjust the prediction distribution using prior information derived from a KB. An overview of the framework is depicted in Figure 4.

3.1 Knowledgeable Pre-Training

This section describes three knowledge-aware self-supervised learning tasks designed to infuse factual knowledge into LLMs, namely, *Masked Entity Prediction* (MEP), *Entity Description Generation* (EDG), and *Knowledgeable Question Answering* (KQA). Differing from Chen et al. (2022a), we leverage an external KB to enrich the models' language generation abilities with respect to important entities. The input consists of a training corpus $\{X\}$ and a KB $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ denotes a set of entities, $\mathcal{R}$ a set of relations, and $\mathcal{T}$ a set of triples representing factual knowledge.

Masked Entity Prediction (MEP). MEP requires the model to predict missing entities within a text, enhancing its capability to learn explicit knowledge. This task is akin to *Masked Language Modeling* employed in BERT-style models (Devlin et al., 2019; Liu et al., 2019). Given a text composed of tokens $X = \{x_i\}$, we identify all entities $E_X = \{e | e \in \mathcal{G}, e \in X\}$ using an entity linking toolkit. Each entity $e = \{x_j | x_j \in X\}$, which may span multiple tokens, is either replaced with special tokens (e.g., "_") or random tokens with equal probability. This process generates a modified text $\hat{X} = \{\hat{x}_i\}$. A label mask vector $\mathcal{M}_{\hat{X}}$ is created to indicate training positions, where $\mathcal{M}_{\hat{X}_i} = \mathbb{I}(\hat{x}_i \in E_X)$ and $\mathbb{I}(\cdot)$ is an indicator function. Figure 4 (left) illustrates this with highlighted words.

Entity Description Generation (EDG). EDG tasks the model with producing a text description for a given entity. For a text X and associated entity set E_X, we construct a prefix text using the template "Entities:", followed by a list of entities and the template "Text:". The original text X serves as the suffix. This forms the modified example $\hat{X}$ and corresponding label mask vector $\mathcal{M}_{\hat{X}}$, where $\mathcal{M}_{\hat{X}_i} = 1$ if $\hat{x}_i$ is part of the suffix string.

Knowledgeable Question Answering (KQA). KQA leverages relation triples from the KB to facilitate question answering. Given a text X and entity set E_X, we select a pair of entities $e_h, e_t \in E_X$ linked by a 1-hop relation $r \in \mathcal{R}$ to form a triple $(e_h, r, e_t) \in \mathcal{T}$. Inspired by Wang et al. (2022), we create a question template for each triple, prompting the model to predict the tail entity e_t. Training examples $\hat{X}$ and label mask vectors are generated accordingly, with $\mathcal{M}_{\hat{X}_i} = 1$ designating tokens belonging to the tail entity.

During pre-training, we randomly compile examples from the same task into a training batch $\mathcal{X} = \{\hat{X}\}$ until the maximum sequence length is reached. The cross-entropy loss for prediction positions (where $\mathcal{M}_{\hat{X}} = 1$) is computed as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{X}|} \sum_{\hat{X} \in \mathcal{X}} \frac{1}{T_{\hat{X}}} \sum_{\hat{x}_i \in \hat{X}} \mathcal{M}_{\hat{X}_i} \log p(y_i | \hat{X}_{<i}), \quad (1)$$

where y_i is the ground truth token, $p(\cdot)$ is the predicted probability, and $T_{\hat{X}} = \sum_{\hat{x}_i \in \hat{X}} \mathcal{M}_{\hat{X}_i}$ is the number of tokens the model is required to predict.

3.2 Knowledgeable Example Retrieval

Despite having a powerful and knowledgeable LLM at our disposal, the efficacy of ICL is significantly influenced by the selection and ordering of labeled examples (Brown et al., 2020). Previous studies (Liu et al., 2022; Lu et al., 2022; Rubin et al., 2022) have demonstrated that LLMs can autonomously generate suitable text-based prompts, yet they largely overlook the importance of *factual knowledge* from KBs. To address this gap, we introduce a novel Knowledgeable Example Retrieval (KER) algorithm that utilizes knowledge to select the most relevant in-context examples. This process is illustrated in Figure 4 (middle) and detailed in Algorithm 1 in Appendix C. Concisely, given a training set $D_{trn} = \{(X_i^{trn}, y_i^{trn}, E_i^{trn})\}$ and a testing set $D_{tgt} = \{(X_j^{tgt}, E_j^{tgt})\}$, where X_i^{trn} and X_j^{tgt} are input texts, y_i^{trn} are labels, and E_i^{trn} and E_j^{tgt} are the corresponding entity sets, KER's objective is to select a subset of training examples as demonstrations that exhibit high knowledge relevance to the testing set.

A straightforward approach is to retrieve examples containing entities that *cover* a higher number of target examples. We use the Jaccard similarity to assess the similarity between two examples:

$$d_{jac}(i,j) = \frac{|E_i^{trn} \cap E_j^{tgt}|}{|E_i^{trn} \cup E_j^{tgt}|}. \tag{2}$$

However, since the Jaccard similarities for most example pairs are zero, we further employ pre-trained knowledge embeddings to retrieve training examples that are semantically *similar* to the target set. We compute the average representations $\mathbf{e}_i$ and $\mathbf{e}_j$ of all entities in E_i^{trn} and E_j^{tgt}, respectively. The semantic difference is quantified using the Euclidean distance $d_{sem}(i,j)$ between $\mathbf{e}_i$ and $\mathbf{e}_j$. The overall knowledge relevance between two examples is calculated as follows:

$$d(X_i^{trn}, X_j^{tgt}) = \alpha \frac{d_{jac}(i,j) + \gamma}{\max_{X_k^{trn} \in \mathcal{D}_{trn}} d_{jac}(i,k) + \gamma} + (1-\alpha)(1 - \frac{d_{sem}(i,j)}{\max_{X_k^{trn} \in \mathcal{D}_{trn}} d_{sem}(i,k)}), \tag{3}$$

where $\alpha \in [0,1]$ and $\gamma > 0$ are tunable hyperparameters. The sampling weight for each training example X_i^{trn} is given by:

$$s'(X_i^{trn}) = \frac{s(X_i^{trn})}{\sum_{X_j^{trn} \in \mathcal{D}_{trn}} s(X_j^{trn})}, \tag{4}$$

where $s(X_i^{trn})$ is computed as the average relevance score to the testing set:

$$s(X_i^{trn}) = \frac{1}{|\mathcal{D}_{tgt}|} \sum_{X_j^{tgt} \in \mathcal{D}_{tgt}} d(X_i^{trn}, X_j^{tgt}). \tag{5}$$

An example with a higher weight signifies greater knowledge relevance across all target examples. Ultimately, we sample K training examples based on these weights to serve as in-context examples.

3.3 Knowledgeable Prediction Calibration

Following model pre-training and in-context example selection, we can proceed to generate predictions for the target example $X^{tgt} \in \mathcal{D}_{tgt}$ using the following equation:

$$\hat{y} = \arg\max_{v \in \mathcal{V}} p(y = e | X, X^{tgt}), \tag{6}$$

where $\mathcal{V}$ is a verbalizer that maps label words to their corresponding classes [5]. $\widetilde{\mathcal{D}}$ represents the set of in-context examples used for prediction. However, as discussed in Section 2, the frequency of label words (in classification tasks) or entities (in question answering tasks) can bias the prediction probabilities. To mitigate this issue, we utilize the prior information of label words to refine the prediction for each target example.

Specifically, we select a subset of training data $\mathcal{S}$ from the KQA task and estimate the contextual prior probability for each candidate label word or entity $v \in \mathcal{V}$ at the output position:

$$P(v) \approx \frac{1}{|\mathcal{S}|} \sum_{\hat{X} \in \mathcal{S}} p(y = v | \hat{X}), \tag{7}$$

where $\hat{X}$ denotes a training example, and $P(v)$ represents the estimated prior probability of candidate v. Following this, we discard any label word or entity v whose prior probability falls below a specific threshold (Hu et al., 2022).

Consequently, we enhance the final output by applying calibrated prediction:

$$\hat{y} = \arg\max_{v \in \mathcal{V}} \frac{p(y = v | \widetilde{\mathcal{D}}, X^{tgt})}{P(v)}. \tag{8}$$

Remarks. While most related works (Hu et al., 2022; Zhao et al., 2021) concentrate on prediction calibration, our approach distinguishes itself by

[5] For classification tasks, $\mathcal{V}$ is the set of label words; for question answering tasks, $\mathcal{V}$ is the entire vocabulary.

leveraging a priori knowledge from a large-scale corpus to debias outputs. This contrasts with methods that rely solely on in-domain data or utilize task-agnostic, content-free inputs (e.g., "N/A").

4 Experiments

4.1 Implementation Settings and Baselines

For the pre-training corpus, we use Wikipedia Dumps (2020/03/01)[6], which consists of 25,933,196 sentences. Further, the KB we used is WikiData5M (Wang et al., 2021b), which includes $3,085,345$ entities and 822 relation types. By default, we choose GPT-2 (large) with 0.8B parameters as the backbone. For downstream tasks, we consider 8 text classification tasks and 4 question answering tasks. The details of corpora and downstream benchmarks are shown in Appendix B. The implementation details of pre-training, prompting, and prediction can be found in Appendix C.

We consider the following baselines: 1) **In-Context Learning (ICL)** is the vanilla version proposed by GPT-3. 2) **Calibrate Before Use (CBU)** (Zhao et al., 2021) is a typical method that aims to de-bias the prediction via content-free prompts. 3) **KATE** (Liu et al., 2022) uses the CLS embeddings of a RoBERTa-large model as sentence representations, and retrieves the nearest K neighbors for each target example as the final in-context examples. 4) **MetaICL** (Min et al., 2022a) improves ICL by meta-learning the objective of ICL in cross-task settings. 5) **SelfSup.** (Chen et al., 2022a) improves ICL by multiple self-supervised learning tasks. We also choose RoBERTa-large to perform fully **Fine-tuning** to demonstrate the ceiling performance of each task.

4.2 Main Results

Table 1 and Table 2 respectively report the results over text classification and question answering tasks in the 8-shot setting. We thus make the following observations: 1) Our proposed framework outperforms strong baselines and achieves substantial improvements over all benchmarks. Specifically, compared with ICL, the average result over the text classification task is improved by 13.70%, which is larger than that of other baselines. The average gain over question answering tasks is also more than 7%, although there is still room for improvement on unseen target domains, likely because they require more challenging generalization and commonsense abilities. 2) Compared with ICL, KER and KCP make significant contributions to the performance. Particularly, KER and KCP also respectively outperform strong baselines KATE and CBU, indicating the indispensable merit of factual knowledge at the inference stage. 3) The performance of KPT exceeds that of meta-learning (MetaICL) and self-supervised learning (SelfSup.) approaches by around 4%, which also focus on continual pre-training. This demonstrates that explicitly injecting knowledge into LLMs is more effective for ICL, which is imperative and plays a dominant role. 4) Our method attains more impressive performance when combining all of these knowledgeable techniques, highlighting the necessity of factual knowledge in ICL. We provide a detailed analysis in Section 4.3. 5) We also evaluate other scales for GPT-2 and OPT in 8-shot settings. Results in Appendix F show that the improvements are consistent across different LLMs.

4.3 Ablation Study

We further investigate how these proposed knowledgeable techniques contribute to the final performance with different combinations. As shown in Table 3, the results demonstrate that any combination greatly promotes the overall performance of vanilla ICL. An interesting observation is that KPT is particularly important for performance improvement, achieving higher scores than KER and KCP. This indicates that the most effective way to unleash the power of LLMs is to inject knowledge into the model parameters. Nonetheless, the combination of KER and KCP also improves ICL by about 8% for each task, respectively. This suggests that KER and KCP are critical to ICL because ultra-large LLMs cannot be continuously pre-trained or tuned in real-world scenarios to save computational resources. Furthermore, results from Table 1 to Table 3 show that our method has significantly improved classification tasks. We believe that the benefits of injecting knowledge are more pronounced for simple language understanding tasks than for question answering.

4.4 Further Analysis

Effectiveness of KPT. To investigate what makes a high performance for KPT, we test the effectiveness of each knowledgeable self-supervised task. For a fair comparison, we also choose two baselines: 1) **None** is that we do not use any self-

[6] https://dumps.wikimedia.org/enwiki/

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	76.18±7.2	54.46±2.3	56.85±2.4	52.93±3.2	53.94±5.0	42.50±1.8	51.56±4.1	45.67±6.6	54.26
CBU (Zhao et al., 2021)	82.71±4.4	63.07±3.9	57.93±2.8	53.19±3.9	54.87±2.8	51.34±1.7	54.61±3.7	55.42±2.8	59.14
KATE (Liu et al., 2022)	81.33±3.8	58.04±3.9	59.40±2.4	53.57±3.5	53.17±2.7	45.48±2.1	54.69±2.8	50.28±3.4	57.00
MetaICL† (Min et al., 2022a)	87.40±5.0	62.91±2.0	60.22±3.4	55.18±1.9	57.06±2.8	49.20±2.5	56.09±1.8	55.80±2.4	60.48
SelfSup.† (Chen et al., 2022a)	87.94±3.0	62.33±2.0	62.00±2.2	54.77±1.8	57.27±2.6	45.80±2.5	55.59±2.5	57.44±3.2	60.39
KICT†	**91.21±2.9**	**69.96±0.7**	**69.59±1.0**	**60.66±1.2**	**63.74±4.2**	**56.07±3.8**	**63.52±5.5**	**68.89±5.7**	**67.96**
only w. KPT†	90.04±3.5	66.65±1.9	67.39±2.6	58.97±3.0	58.26±3.3	55.43±2.0	60.16±2.2	59.74±4.4	64.58
only w. KER	84.05±2.7	59.26±2.5	59.93±1.0	57.23±1.2	53.79±4.0	51.36±3.8	55.52±5.1	52.70±3.3	59.23
only w. KPC	85.52±3.9	64.77±0.7	63.13±1.2	57.69±2.4	55.94±1.2	54.07±2.8	56.92±2.7	57.24±5.5	61.91

Table 1: The 8-shot performance (%) on GPT-2 (large) of different learning settings with standard deviations over text classification benchmarks. Compared with other baselines, our framework achieves consistent improvement. † denotes the method involves parameters update for ICL. "only w." means we only use one technique in KICT.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	27.93±4.8	54.49±3.5	46.93±3.0	40.31±2.7	42.42
CBU (Zhao et al., 2021)	29.88±3.9	55.40±1.8	49.32±4.0	44.05±4.0	44.66
KATE (Liu et al., 2022)	29.02±4.0	55.10±3.9	47.25±3.4	42.77±3.8	43.54
MetaICL† (Min et al., 2022a)	31.16±3.2	55.64±2.9	50.46±2.6	46.72±2.7	46.00
SelfSup.† (Chen et al., 2022a)	31.32±3.0	54.88±3.0	49.97±2.7	47.50±3.5	45.92
KICT†	**36.17±1.8**	**58.11±2.4**	**54.23±2.6**	**50.46±3.3**	**49.74**
only w. KPT†	34.21±4.3	57.32±2.2	52.79±3.0	49.93±1.9	48.56
only w. KER	29.56±2.3	55.82±1.2	48.11±2.4	43.58±2.1	44.27
only w. KCP	33.60±3.7	57.77±2.4	51.63±2.9	46.09±3.1	47.27

Table 2: The 8-shot performance (%) on GPT-2 (large) of different learning settings with standard deviations over question answering benchmarks.

supervised task, which is the same as vanilla ICL proposed in (Brown et al., 2020), 2) **GPT-2** represents conventional autoregressive language modeling (ALM) pre-training tasks. As shown in Table 4, KPT can make substantial improvements for ICL. Particularly, all the self-supervised learning tasks in KPT are complementary for pre-training and outperform the baseline with or without the conventional objective of GPT-2. In addition, the MEP and KQA tasks are most critical for classification and question answering, respectively, which demonstrates that different pre-training objectives possess different advantages in downstream tasks.

Sample Effectiveness. To investigate the influence of the number of in-context examples K, we choose multiple classification and question answering tasks and vary K from 0, 1, 4, 8 to 16. From Figure 5, we find that increasing K generally helps across both classification and question answering tasks, demonstrating that more in-context examples may bring more knowledge to better guide the LLM to make predictions. When $K > 8$, the performance of the most tasks will decrease, because the maximum length limit causes information loss. The suitable value K is set around 8.

Visualization of Selected Examples in KER. In addition, for explicitly seeing the performance in semantic space, we obtain the t-SNE (Van der Maaten and Hinton, 2008) visualization of each training example over AGNews via averaged representations of all corresponding entities. We choose KATE as our strong baseline, which is also focused on the example selection. Here, we do not fine-tune RoBERTa on the training set. Figure 6 demonstrates that our method can build better semantic representations toward factual knowledge.

Permutations of In-Context Examples. We also compare different permutations of these selected

Baselines	SST-2 acc	MRPC f1	MNLI acc	RTE acc	AGNews acc	TREC acc	ComQA acc	Quartz acc	SQuAD em	Quoref em
ICL	76.18±7.2	54.46±2.3	56.85±2.4	53.94±5.0	45.67±6.6	51.56±4.1	27.93±4.8	54.49±3.5	46.93±3.0	40.31±2.7
KPT+KER	91.04±3.3	67.93±3.0	68.47±2.9	61.30±3.3	62.18±3.9	61.52±3.1	35.17±4.0	57.64±2.6	52.23±3.4	50.20±3.1
KPT+KCP	90.65±3.7	68.44±2.5	68.89±3.4	62.38±2.3	63.88±3.5	62.12±2.9	**36.38±2.2**	58.03±2.0	54.17±1.8	50.18±2.2
KER+KCP	86.45±3.0	64.07±2.4	66.60±2.9	57.39±3.2	58.95±3.6	58.60±3.5	34.26±2.2	57.88±3.1	52.20±2.3	47.92±2.7
All (KICT)	**91.21±2.9**	**69.96±0.7**	**69.59±1.0**	**63.74±4.2**	**68.89±5.7**	**63.52±5.5**	36.17±1.8	**58.11±2.4**	**54.23±2.6**	**50.46±3.1**

Table 3: The 8-shot performance (%) of different combinations of the knowledgeable modules.

Methods	SST-2 acc	AGNews acc	TREC acc	ComQA acc	SQuAD em
None (ICL)	76.18±7.2	45.67±6.6	51.56±4.1	27.93±4.8	46.93±3.0
GPT-2	81.35±3.0	48.72±2.7	52.36±3.3	28.61±3.8	47.14±3.1
KPT	**90.04±3.5**	**59.74±4.4**	**60.16±2.0**	**34.21±4.3**	**52.79±3.0**
w/o. MEP	84.40±4.0	51.29±3.9	54.72±3.1	33.01±7.7	52.23±2.8
w/o. EDG	87.19±2.9	56.40±4.3	55.91±3.1	31.95±5.9	50.80±3.9
w/o. KQA	85.30±3.3	53.03±3.6	53.46±2.4	30.08±5.8	49.71±4.6

Table 4: The 8-shot performance (%) of each self-supervised task. GPT-2 denotes the vanilla objective.

Baselines	SST-2	MRPC	MNLI
Random	79.42±2.7	**59.26±2.5**	**59.93±1.0**
Ascending	78.29±2.2	58.05±2.6	59.31±1.5
Descending	**79.61±3.0**	58.16±3.0	59.58±1.3

Table 5: The 8-shot averaged results (%) of KICT (only w. KER) for different permutations.

examples according to the sample weight computed in Eq. 4. In Table 5, Random means to randomly choose an order. Ascending and Descending respectively denote that the example order is ascending or descending by weight. From the results, we find no tangible relationship between the sampling weight and order.

Effectiveness of KPC. We finally conduct analysis on prediction calibration. We choose AGNews and TREC tasks and follow the same settings in the preliminary experiments (we randomly choose two label words from different frequency regions). Results in Figure 7 demonstrate that calibrating the prediction consistently achieves improvements to the vanilla approach. In addition, we find that the prediction results highly depend on the label frequency, which is similar to Figure 3. However, our KPC still outperforms the strong baseline Calibrate Before Use (CBU) with arbitrary label frequency, which only transforms the input into content-free prompts. It underscores that the prior information of each label word in KB is non-negligible. In other words, calibration by the prior information can alleviate the impact of label frequency.

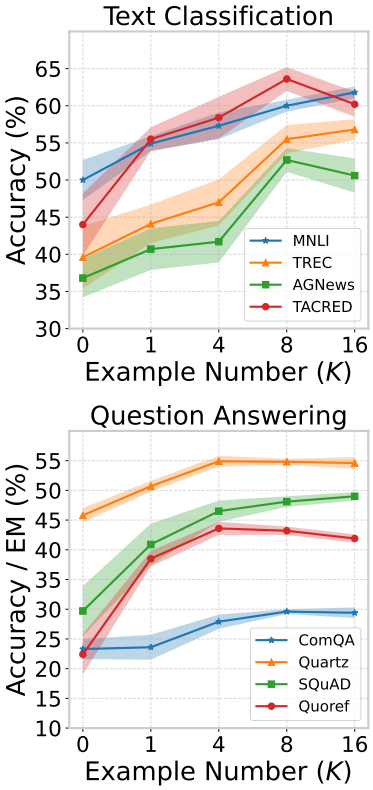


Figure 5: GPT-2 (large) sample effectiveness (%) of KICT (only w. KER) with different values of K.

5 Related Work

5.1 Pre-trained/Large Language Models

Pre-trained Language Models (PLMs) aim to learn representations from texts and have made significant progress in NLP. PLMs can be divided into three main categories: encoder-only (Devlin et al., 2019; Liu et al., 2019; He et al., 2021; Yang et al., 2019; Lan et al., 2020; Zhang et al., 2022b), decoder-only (Radford et al., 2018; Brown et al., 2020; Zhang et al., 2022a), and encoder-decoder (Lewis et al., 2020; Raffel et al., 2020). To incorporate factual knowledge into PLMs, a branch of knowledge-enhanced PLMs has been proposed (Zhang et al., 2019; Sun et al., 2020a; Wang et al., 2021b,a, 2022; Pan et al., 2022; Zhang et al.,

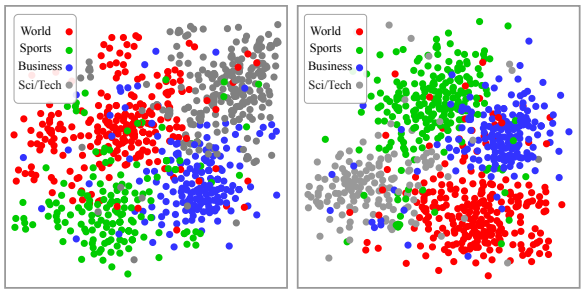


Figure 6: Visualizations of each AGNews's training example. KATE (left) uses CLS embeddings of RoBERTa. Ours (right) utilizes averaged knowledge embeddings.

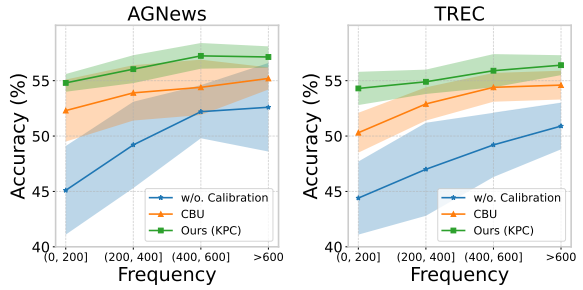


Figure 7: GPT-2 (large) 4-shot performance of calibration over difference word frequencies.

2022c), enabling PLMs to capture rich semantic knowledge from KBs. Since the introduction of ChatGPT, a variety of decoder-only LLMs have been released. Popular open-source LLMs include LLaMA (Touvron et al., 2023), OPT (Zhang et al., 2022a), Galactica (Taylor et al., 2022), Pythia (Biderman et al., 2023), among others. Our work concentrates on decoder-only LLMs and aims to infuse them with factual knowledge to enhance their ICL performance.

5.2 Prompt Learning

Prompt-based learning aims to add natural language prompts to guide PLMs to solve downstream tasks. A series of works focus on tunable discrete prompt tuning (Gao et al., 2021; Raffel et al., 2020) and continuous prompt tuning (Liu et al., 2021b; Gu et al., 2021; Xu et al., 2023). For LLMs, GPT-3 (Brown et al., 2020) enables In-Context Learning (ICL) with a text-based prompt in zero-shot scenarios, bypassing parameter updates (Dong et al., 2023). To explore the factors affecting ICL, previous works have focused on input-output mapping (Min et al., 2022b; Kim et al., 2022), meta-learning (Chen et al., 2022b; Min et al., 2022a), prompt engineering (Liu et al., 2022, 2021a), and prediction calibration (Zhao et al., 2021; Hu et al., 2022), among others. Recently, the Chain-of-Thought (CoT) approach has been presented to leverage reasoning and interpretable information to guide LLMs in generating reliable responses (Si et al., 2022; Zhang et al., 2022d; Wei et al., 2022; Yan et al., 2023). Different from these approaches, we exploit *factual knowledge* to further improve ICL in pre-training, prompting, and prediction phases.

6 Conclusion

In this paper, we investigate and harness *factual knowledge* in ICL, including inherent knowledge embedded in LLMs, pertinent knowledge derived from selected training examples, and knowledge biases affecting predictions. We introduce a novel Knowledgeable In-Context Tuning (**KICT**) framework to further enhance ICL performance by comprehensively exploiting factual knowledge throughout the processes of pre-training, prompting, and prediction. Experiments demonstrate that each introduced technique significantly improves upon strong baselines across classification and question-answering tasks. Future work will focuses on 1) exploring the reasoning capabilities and interpretability of knowledge within ICL, and 2) extending our approach to encoder-decoder models.

Acknowledgements

This work has been supported by the National Natural Science Foundation of China under Grant No. U1911203, Alibaba Group through the Alibaba Innovation Research (AIR) Program, the National Natural Science Foundation of China under Grant No. 61877018, the Research Project of Shanghai Science and Technology Commission (20dz2260300) and The Fundamental Research Funds for the Central Universities.

Limitations

This work presents several limitations: 1) It concentrates on decoder-only LLMs, as traditional in-context learning primarily targets decoder-only generation models such as GPT-2, GPT-3, OPT, etc. Nevertheless, we envision potential extensions to encoder-decoder architectures used in tasks such as translation and conditional generation. 2) Due to computational resource constraints, we do not experiment with ultra-large LLMs exceeding 10 billion parameters. 3) Our investigation centers on factual knowledge in three specific areas: pre-training,

prompting, and prediction. We acknowledge that knowledge may influence additional aspects such as reasoning and interpretability, and we intend to explore these in future research.

Ethical Considerations

The contributions of this work are methodological, focusing on a Knowledgeable In-Context Tuning (**KICT**) framework to augment the capabilities of LLMs with factual knowledge. Nonetheless, transformer-based models may perpetuate negative biases, including gender and social biases. As such, these issues are inherent to our work as well. We advise caution and recommend addressing potential risks when **KICT** models are deployed in real-world applications.

References

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

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

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022a. Improving in-context few-shot learning via self-supervised training. In *NAACL*, pages 3558–3573.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In *EMNLP*, pages 15651–15662.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022b. Meta-learning via language model in-context tuning. In *ACL*, pages 719–730.

Pradeep Dasigi, Nelson F. Liu, Ana Marasovic, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *EMNLP*, pages 5924–5931.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL*, pages 3816–3830.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. PPT: pre-trained prompt tuning for few-shot learning. *CoRR*, abs/2109.04332.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *ICLR*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *ACL*, pages 2225–2240.

Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *CoRR*, abs/2205.12685.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. In *EMNLP*, pages 4329–4343. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *ACL*, pages 100–114.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *AAAI*, pages 2901–2908.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: zero-shot in-context learning with pseudo-demonstrations. In *ACL*, pages 2304–2317.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. Metaicl: Learning to learn in context. In *NAACL*, pages 2791–2809.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *CoRR*, abs/2202.12837.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, pages 3470–3487.

Xiaoman Pan, Wenlin Yao, Hongming Zhang, Dian Yu, Dong Yu, and Jianshu Chen. 2022. Knowledge-in-context: Towards knowledgeable semi-parametric language models. *CoRR*, abs/2210.16433.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*, pages 784–789.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *NAACL*, pages 2655–2671. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan L. Boyd-Graber, and Lijuan Wang. 2022. Prompting GPT-3 to be reliable. *CoRR*, abs/2210.09150.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020a. Colake: Contextualized language and knowledge embedding. In *COLING*, pages 3660–3670.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020b. ERNIE 2.0: A continual pre-training framework for language understanding. In *AAAI*, pages 8968–8975.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. In *EMNLP*, pages 5940–5945.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*, pages 4149–4158.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa,

Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR*, pages 200–207. ACM.

Jianing Wang, Nuo Chen, Qiushi Sun, Wenkang Huang, Chengyu Wang, and Ming Gao. 2023a. Hugnlp: A unified and comprehensive library for natural language processing. In *CIKM*, pages 5111–5116. ACM.

Jianing Wang, Wenkang Huang, Qiuhui Shi, Hongbin Wang, Minghui Qiu, Xiang Li, and Ming Gao. 2022. Knowledge prompting in pre-trained language model for natural language understanding. *CoRR*, abs/2210.08536.

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. 2023b. Boosting language models reasoning with chain-of-knowledge prompting. *CoRR*, abs/2306.06427.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *ACL*, pages 1405–1418.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *TACL*, 9:176–194.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023c. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Zhenyu Wu, Yaoxiang Wang, Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Jingjing Xu, and Yu Qiao. 2023. Openicl: An open-source framework for in-context learning. In *ACL*, pages 489–498.

Ziyun Xu, Chengyu Wang, Minghui Qiu, Fuli Luo, Runxin Xu, Songfang Huang, and Jun Huang. 2023. Making pre-trained language models end-to-end few-shot learners with contrastive prompt tuning. In *WSDM*, pages 438–446. ACM.

Junbing Yan, Chengyu Wang, Taolin Zhang, Xiaofeng He, Jun Huang, and Wei Zhang. 2023. From complex to simple: Unraveling the cognitive tree for reasoning with small language models. In *EMNLP (Findings)*, pages 12413–12425. Association for Computational Linguistics.

Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. 2021. A survey of knowledge enhanced pre-trained models. *CoRR*, abs/2110.00269.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Taolin Zhang, Junwei Dong, Jianing Wang, Chengyu Wang, Ang Wang, Yinghui Liu, Jun Huang, Yong Li, and Xiaofeng He. 2022b. Revisiting and advancing chinese natural language understanding with accelerated heterogeneous knowledge pre-training. In *EMNLP*, pages 560–570. Association for Computational Linguistics.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022c. DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *AAAI*, pages 11703–11711. AAAI Press.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *ACL*, pages 1441–1451.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022d. Automatic chain of thought prompting in large language models. *CoRR*, abs/2210.03493.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *ICLR*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023a. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *ACL*, pages 5823–5840.

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

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Details of Preliminary Experiments

A.1 Details of Destruction Settings

For our preliminary experiments, we selected 8 classification tasks and 4 question-answering tasks. The specifics of these datasets are detailed in Appendix B. To explore the influence of *factual knowledge*, we posit that entities (and their associated labels in text classification tasks) embody factual knowledge (Wang et al., 2021b, 2022, 2021a; Sun et al., 2019; Zhang et al., 2019). We identify all entities using the open-source TagMe entity linking tool[7] (Ferragina and Scaiella, 2010). In the case of classification tasks, labels are treated as special types of entities. We follow the methodologies of Min et al. (2022b) and Kim et al. (2022) to create various destruction settings that either remove or replace entities (and labels), thereby demonstrating the impact of factual knowledge. Additionally, for each task, we randomly select $K = 8$ examples as in-context examples and concatenate them with each test example to form an input sequence, capped at a maximum sequence length of 256 tokens. With 5 different random seeds (i.e., 12, 24, 42, 90, and 100), each dataset yields 5 unique test results for a given LLM. Consequently, for each LLM, we collate $8 \times 5 = 40$ results for classification and $4 \times 5 = 20$ results for question-answering tasks. The aggregated results are presented in Figure 2, underscoring factual knowledge as a pivotal component in the performance of ICL.

[7] https://sobigdata.d4science.org/group/tagme

A.2 Details of Frequency Settings

In our preliminary assessment of label word frequency's impact, we focused on two well-established tasks: AGNews and TREC. Selecting $K = 4$ examples from the training corpus to construct the in-context prompt, we then used the remaining training examples as targets to generate predictions. Development or test sets were not utilized due to their insufficient scale for demonstrating frequency effects clearly. During prediction, we recorded the top-4 words with the highest prediction probabilities, facilitating the computation of frequency statistics for each label word. Figure 8 depicts the top-8 label word frequency statistics for each AGNews category. To examine frequency influences, we randomly selected two label words per frequency range (e.g., $(0, 200]$, $(200, 400]$, $(400, 600]$, and > 600) for predictions. For instance, in AGNews, labels like "teams" and "groups" could be chosen from the > 600 frequency region to represent the "sports" category. Accordingly, we generated $2^4 = 16$ and $2^6 = 64$ permutations for AGNews and TREC, respectively. We report the average results using GPT-2 (urge) with 1.5B parameters and present the findings in box plot format in Figure 3.

A.3 Analysis of Knowledge Relevance in In-Context Examples

Our preliminary experiments indicated that factual knowledge in selected in-context examples is crucial for ICL. To substantiate this, we conducted further analyses on two datasets, SST-2 and TREC. Employing our KER technique, we calculated a knowledge relevance score for each training example. For each defined score interval (i.e., $(0, 15]$, $(15, 30]$, $(30, 45]$, $(45, 60]$, $(60, 75]$), we sampled $K = 4$ examples to compose the in-context prompt. We then assessed the average performance across all $4! = 24$ permutations for each interval and visualized the results in Figure 9. The findings corroborated the significance of selecting examples with high knowledge relevance for enhancing ICL performance.

B Details of the Corpus and Downstream Benchmarks

B.1 Corpora and Knowledge Base

We propose knowledgeable pre-training (KPT), which is similar to the current flourishing research

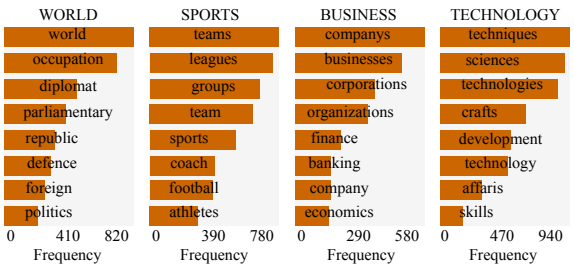


Figure 8: Label word frequency statistics for the AG-News dataset.

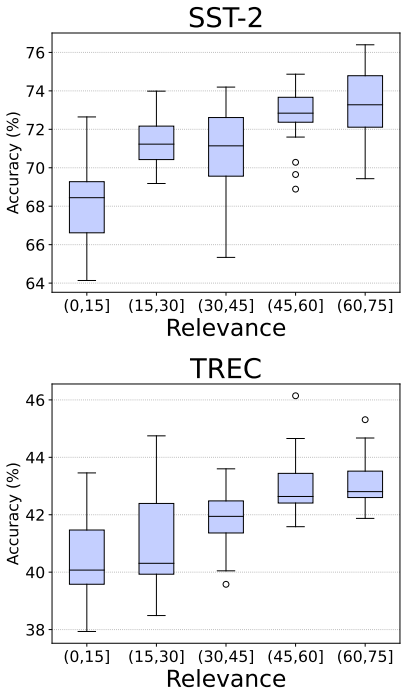


Figure 9: The 4-shot performance (%) with different knowledge relevance over SST-2 and TREC.

of *knowledge-enhanced pre-trained language models* (KEPLMs) (Liu et al., 2020; Sun et al., 2019, 2020b; Wang et al., 2022). Different from them, we focus on auto-regressive PLMs, such as GPT-2. We collect training corpora from Wikipedia (2020/03/01)[8], and use WikiExtractor[9] to process the pre-training data. The knowledge base (KB) $\mathcal{G}$ we choose is WikiData5M (Wang et al., 2021b), which is an urge-large structural data source based on Wikipedia. The entity linking toolkit we used is TagMe. In total, we have 3,085,345 entities and 822 relation types in $\mathcal{G}$, and 25,933,196 training sentences.

As mentioned above, KPT consists of three self-training tasks, i.e., *masked entity prediction*, *entity description generation*, and *knowledgeable question answering*. For each task, we randomly select multiple sentences to form a training instance until reaching the maximum sequence length (i.e., 2048). Finally, we have sampled 100k training instances for each task. In average, we have 8 examples for each instance.

B.2 Downstream Task Datasets

To evaluate the effectiveness of our framework, we choose 8 text classification tasks and 4 question answering tasks. For the text classification, we directly choose 8 tasks from (Gao et al., 2021; Zhao et al., 2021). All the classification tasks involve sentiment analysis, natural language inference (NLI), question classification, and topic classification. For the question answering tasks, we choose four widely used tasks, including CommonsenseQA (ComQA) (Talmor et al., 2019), Quartz (Tafjord et al., 2019), SQuAD (Rajpurkar et al., 2018) and Quoref (Dasigi et al., 2019), where ComQA and Quartz are multi-choice QA, SQuAD and Quoref are extractive QA. The statistics of each dataset are shown in Table 6.

C Implementation Details

C.1 Pre-training Details

In the pre-training stage, we choose different scales of GPT-2 (0.1B, 0.3B, 0.8B, 1.5B) (Brown et al., 2020) and OPT (Zhang et al., 2022a) (2.7B, 6.7B) from HuggingFace[10] as the underlying LLMs. We do not use larger GPT-3 models because of the computation resource limitations. Because all three kinds of pre-training tasks share the same format, we can directly mix up all the pre-training examples to form a cross-task pre-training paradigm. We find that it is suitable for the LLM to learn cross-task knowledge. We train our model by AdamW algorithm with $\beta_1 = 0.9, \beta_2 = 0.98$. The learning rate is set as 1e-5 with a warm-up rate 0.1. We also leverage dropout and regularization strategies to avoid over-fitting. The models are trained on 8 NVIDIA A100-80G GPUs.

C.2 Prompting Details

We describe the implementation details with knowledgeable example retrieval (KER). Given a training

[8] https://dumps.wikimedia.org/enwiki/.
[9] https://github.com/attardi/wikiextractor.
[10] https://huggingface.co/transformers/index.html.

Category	Dataset	#Class	#Train	#Test	Type	Labels (classification tasks)
Text Classification	SST-2	2	6,920	872	sentiment	positive, negative
	MRPC	2	3,668	408	paraphrase	equivalent, not_equivalent
	MNLI	3	392,702	9,815	NLI	entailment, neutral, contradiction
	QNLI	2	104,743	5,463	NLI	entailment, not_entailment
	RTE	2	2,490	277	NLI	entailment, not_entailment
	CB	3	250	57	NLI	entailment, neutral, contradiction
	TREC	6	5,452	500	question cls.	abbr., entity, description, human, loc., num.
	AGNews	4	120,000	7,600	topic cls.	world, sports, business, technology
Question Answering	ComQA	-	9,741	1,221	multi-choice	-
	Quartz	-	2,696	384	multi-choice	-
	SQuAD	-	87,599	10,570	extractive QA	-
	Quoref	-	19,399	2,418	extractive QA	-

Table 6: The statistics of multiple text classification and question answering datasets. Since the original test data is unavailable, we use the development sets as our test sets.

dataset and a testing set, we aim to choose K examples from the training set which have a high knowledge relevant to all testing examples. To reach this goal, we utilize both Jaccard similarity and Euclidean distance in terms of pre-trained knowledge embeddings. For pre-trained knowledge embeddings, we choose the ConVE (Dettmers et al., 2018) algorithm to pre-train over wikidata5m and obtain the embeddings of entities and relations. We set its dimension as 768, the negative sampling size as 64, the batch size as 128 and the learning rate as 0.001. Finally, we only store the embeddings of all the entities. The KER algorithm for the prompting is shown in Algorithm 1.

C.3 Prediction Details

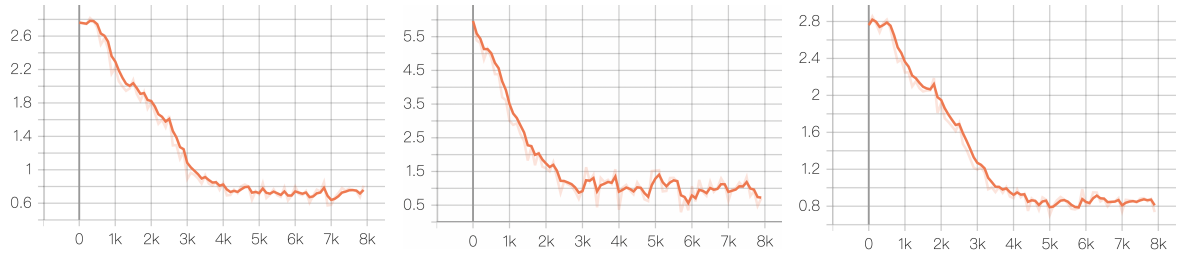
We first provide the details of the prompt formats and label mapping rules. Specifically, for the classification task, we need to define a template and label mapping to guide the model to generate results toward pre-defined classes. The prompt formats and label words are shown in Table 8. For the question answering task, we only need to define the template format, shown in Table 9.

During the prediction, we calibrate the prediction probability. We thus provide the implementation details. We obtain a subset of training corpora from the KQA pre-training task, which consists of many question answer pairs. Thus, for each question, we can generate an answer (may be an entity or a label word) at the output position, and obtain the contextualized prior via Eq. 7. The value $P(v)$ means the prior information of the generated entity or label word. Intuitively, if the value $P(v)$ is higher, the entity or label word v is more likely

Algorithm 1 Knowledgeable Example Retrieval

Require: Training set $\mathcal{D}_{trn}$, Target (testing) set $\mathcal{D}_{tgt}$, number of in-context examples K.

Randomly sampling a subset $\mathcal{D}'_{trn}$ from $\mathcal{D}_{trn}$;
for each target example $(X_j^{tgt}) \in \mathcal{D}_{tgt}$ **do**
  Extract entities E_j^{tgt} from this target example;
  for each training example $(X_i^{trn}, y_i^{trn}) \in \mathcal{D}'_{trn}$ **do**
    Extract entities E_i^{trn} from this training example;
    Calculate Jaccard similarity $d_{jac}(i, j)$ and Euclidean distance $d_{sem}(i, j)$;
  end for
  Conditioning on the target example X_j^{tgt}, obtain the knowledge relevance score $d(X_i^{trn}, X_j^{tgt})$ for the training example X_i^{trn};
end for
Calculate the final sampling weight $s'(X_i^{trn})$ for each training example X_i^{trn} in Eq. 4;
Sampling K training examples via the weight $s'(X_i^{trn})$;
return The selected K training examples.



(a) Masked Entity Prediction (b) Entity Description Generation (c) Knowledge Question Answering

Figure 10: The curves of the pre-training loss on GPT-2 (large) for each self-supervised learning task.

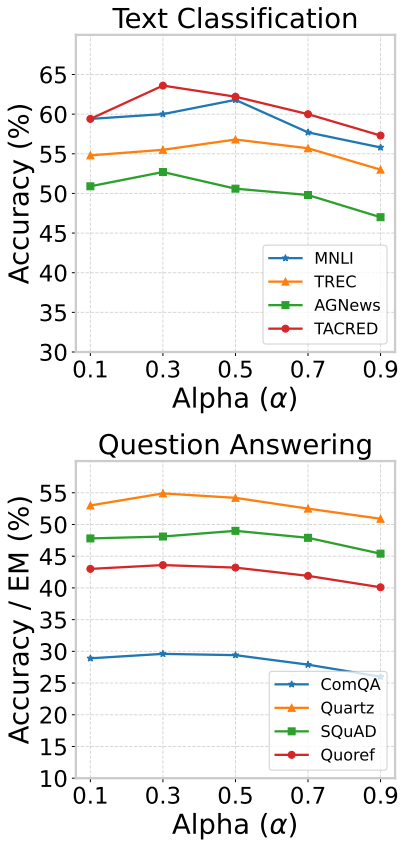


Figure 11: The 8-shot performance (%) of GPT-2 (large) with different α over text classification and question answering tasks.

to be generated. We can save these prior values before prediction for downstream tasks. During the prediction, we can use the prior information of each pre-defined label word or entity to calibrate the prediction probability via Eq. 8.

D Analysis of Settings of Model Variants

We conduct some detailed analysis of our proposed technique.

Analysis of Pre-training Efficiency. To show the efficiency of pre-training, we choose GPT-2 (large) draw the pre-training loss for each self-supervised learning task. From Figure 10, we can see that as the training process proceeds, each self-supervised learning task has reached the convergence of the model through the entire pre-training process.

Hyper-parameter	Value
Batch Size	{2, 4, 8, 16, 32, 64}
Seed	{12, 24, 42, 90, 100}
K	{0, 1, 4, 8, 16}
α	{0.1, 0.3, 0.5, 0.7, 0.9}
γ	{0.001, 0.01, 0.05, 0.1, 0.5, 1.0}

Table 7: The searching scope for each hyper-parameter.

Effectiveness of Hyper-parameters. In KICT, we investigate the effectiveness of the hyper-parameter α in KER, which aims to balance the relevance scores between Jaccard similarity and Euclidean distance. Results shown in Figure 11 demonstrate that the hyper-parameter α is key to the performance. We can see that the suitable value is around 0.3.

Effectiveness of the Template. We believe that the model performances rely on the format of the template, which has been investigated in (Liu et al., 2022; Min et al., 2022b). We choose some other templates for evaluation. For example, when we change the prefix string (e.g., "Question:", "Answer:") to others (e.g., "Q:", "A:"), the performance improvement of KICT is consistent. In addition, we also find that the text split character "\n" between each sentence or example is important to support the generation, which is also found in (Dong et al., 2023; Andrew and Gao, 2007; Kim et al., 2022; Si et al., 2022).

Task	Prompt	Label Words
SST-2	Review: This movie is amazing! Sentiment: Positive Review: Horrific movie, don't see it. Sentiment:	Positive, Negative
MRPC	Whether the two questions are similar? Question 1: How much is this book? Question 2: How many books? Output: No Question 1: Do you know the reason? Question 2: What's the reason? Output:	Yes, No
MNLI	Is entailment, neutral, or contradiction between two texts? Text 1: We sought to identify practices within the past 5 years. Text 2: We want to identify practices commonly used by agencies in the last 5 years. Output: entailment Text 1: yeah well you're a student right Text 2: Well you're a mechanics student right? Output:	entailment, neutral, contradiction
QNLI	Whether the answer is entailed to the question? Text 1: In what year did the university first see a drop in applications? Text2: In the early 1950s, student applications declined as a result of increasing crime and · · · Output: Yes Text1: When did Tesla move to Gospic? Text2: Tesla was the fourth of five children. Output:	Yes, No
RTE	Others argue that Mr. Sharon should have negotiated the Gaza pullout - both to obtain at least some written promises of · · · Question: Mr. Abbas is a member of the Palestinian family. True or False? Answer: False The program will include Falla's "Night in the Gardens of Spain," Ravel's Piano · · · Question: Beatrice and Benedict is an overture by Berlioz. True or False? Answer:	True, False
CB	But he ended up eating it himself. I was reluctant to kiss my mother, afraid that somehow her weakness and unhappiness would infect me. · · · Question: her life and spirit could stimulate her mother. True, False, or Neither? Answer: Neither Valence the void-brain, Valence the virtuous valet. Why couldn't the figger choose his own portion of titanic anatomy to shaft? Did he think he was helping? Question: Valence was helping. True, False, or Neither? Answer:	True, False, Neither
TREC	Classify the questions based on whether their answer type is a Number, Location, Person, Description, Entity, or Abbreviation. Question: How did serfdom develop in and then leave Russia? Answer Type: Description Question: When was Ozzy Osbourne born? Answer Type:	Number, Location, Person, Description, Entity, Abbreviation
AGNews	Article: USATODAY.com - Retail sales bounced back a bit in July, and new claims for jobless benefits fell last week, the government said Thursday, indicating · · · Answer: Business Article: New hard-drive based devices feature color screens, support for WMP 10. Answer:	World, Sports, Business, Technology

Table 8: The prompts used for text classification. We show one training example per task for illustration purposes. The right column shows the label words (aiming to map the word to the original label class).

Task	Prompt
ComQA	Answer the question through multiple-choice. Question: When people want to watch a new move, the often go see it at the? (A) town (B) conference (C) bathroom (D) theater (E) train station Answer: theater Question: Where is known to always have snow? (A) africa (B) north pole (C) roof (D) canada (E) surface of earth north pole Answer:
Quartz	Answer the question through multiple-choice. Question: Eric pushes an electron closer to the nucleus of an atom. The electron _____ energy.As you go farther from the nucleus of an atom, the electron levels have more and more energy. (A) loses (B) gains Answer: gains Question: When something is very lightweight what does it need to move?Objects with greater mass have greater inertia. (A) more inertia (B) less inertia Answer:
SQuAD	Read the question and find an answer in the context. Question: Where was the first figure skating championship held? Context: The tourism industry began in the early 19th century when foreigners visited the Alps, traveled to the bases of the mountains to enjoy the scenery, and stayed at the spa-resorts. Large hotels were built during the Belle Époque; cog-railways, built early in the 20th century, brought tourists to ever higher elevations, with the Jungfraubahn terminating at the Jungfraujoch, well above the eternal snow-line, after going through a tunnel in Eiger. During this period winter sports were slowly introduced: in 1882 the first figure skating championship was held in St. Moritz, and downhill skiing became a popular sport with English visitors early in the 20th century, as the first ski-lift was installed in 1908 above Grindelwald. Answer: St. Moritz Question: What are some examples of classical violinists from Portugal? Context: In the classical music domain, Portugal is represented by names as the pianists Artur Pizarro, Maria João Pires, Sequeira Costa, the violinists Carlos Damas, Gerardo Ribeiro and in the past by the great cellist Guilhermina Suggia. Notable composers include José Vianna da Motta, Carlos Seixas, João Domingos Bomtempo, João de Sousa Carvalho, Luís de Freitas Branco and his student Joly Braga Santos, Fernando Lopes-Graça, Emmanuel Nunes and Sérgio Azevedo. Similarly, contemporary composers such as Nuno Malo and Miguel d'Oliveira have achieved some international success writing original music for film and television. Answer:
Quoref	Read the question and find an answer in the context. Question: What's the name of the person whose birth causes Sarah to die? Context: Jack and Sarah are expecting a baby together, but a complication during the birth leads to the death of Sarah. Jack, grief-stricken, goes on an alcoholic bender, leaving his daughter to be taken care of by his parents and Sarah's mother, until they decide to take drastic action: they return the baby to Jack whilst he is asleep, leaving him to take care of it. · · · Answer: Sarah Question: What is the first name of the person the actor believes is a little too odd? Context: When a British secret agent is murdered in the line of duty, agent Karen Bentley inherits the mission from her partner. The mission is to deliver a flight plan for a hundred American bomber planes to a British agent in Chicago. The plans are hidden in a small medallion of a scorpion that Karen wears. · · · Answer:

Table 9: The prompts used for question answering. We show one training example per task for illustration purposes.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	66.58±4.7	44.73±2.5	49.80±2.9	46.33±2.2	45.70±3.8	36.92±2.3	44.38±2.6	40.53±4.0	46.87
CBU (Zhao et al., 2021)	74.19±4.1	48.88±3.3	51.10±2.5	48.39±3.2	40.07±3.0	39.26±2.8	47.94±2.2	43.28±2.2	49.14
KATE (Liu et al., 2022)	72.38±2.9	46.38±3.2	49.15±3.0	47.28±2.8	46.30±2.6	41.48±2.1	47.80±2.2	43.83±3.1	49.95
MetaICL† (Min et al., 2022a)	77.20±3.6	51.21±2.5	53.29±3.0	49.42±2.2	48.33±2.0	40.18±1.9	49.68±2.8	47.35±2.9	52.08
SelfSup.† (Chen et al., 2022a)	78.94±3.0	52.13±2.0	52.70±2.2	48.29±1.8	49.27±2.6	41.80±2.5	48.59±2.5	47.39±3.2	52.39
KICT†	**82.18±3.2**	**54.19±3.7**	**54.85±2.3**	**50.93±1.9**	**50.13±2.2**	**43.89±2.8**	**51.38±2.5**	**51.20±3.0**	**54.90**

Table 10: The 8-shot performance (%) on GPT-2 (small) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	71.39±3.2	49.60±2.8	53.90±2.4	50.04±3.2	51.18±4.1	39.33±2.8	49.20±2.1	43.75±3.6	51.05
CBU (Zhao et al., 2021)	77.71±3.8	55.48±3.1	55.41±2.2	51.10±3.0	47.53±2.8	48.11±2.7	51.52±2.7	53.27±2.4	55.02
KATE (Liu et al., 2022)	75.32±3.1	53.80±3.1	48.88±3.4	50.14±2.5	45.82±2.9	47.05±2.4	50.25±2.8	51.93±3.4	52.89
MetaICL† (Min et al., 2022a)	80.16±3.0	61.33±2.0	56.12±3.1	54.24±2.9	54.93±2.9	46.50±2.9	53.22±2.8	53.36±2.4	57.48
SelfSup.† (Chen et al., 2022a)	81.62±3.0	58.43±3.2	59.53±2.6	51.70±3.8	54.33±2.6	43.48±3.5	53.46±2.6	53.73±3.1	57.04
KICT†	**89.10±3.9**	**66.44±2.7**	**64.85±3.0**	**57.81±3.2**	**61.02±4.0**	**53.91±2.3**	**60.34±2.0**	**61.77±3.3**	**64.41**

Table 11: The 8-shot performance (%) on GPT-2 (medium) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

E Details of the Grid Search

For the downstream task inference, the searching scope of each model hyper-parameter is shown in Table 7.

F Performance on Different LLMs

To show that our method is general and can be applied to other similar models, we choose other scale sizes of GPT-2 and OPT to show the effectiveness of our KICT. More other experiments results are shown from Table 10 to Table 17.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	78.98±7.2	56.36±2.3	58.25±2.4	55.03±3.2	55.01±5.0	44.04±1.8	53.29±4.1	47.33±6.6	56.04
CBU (Zhao et al., 2021)	83.31±4.4	65.17±3.9	58.13±2.8	55.59±3.9	55.97±2.8	53.14±1.7	56.29±3.7	57.89±2.8	60.69
KATE (Liu et al., 2022)	82.55±3.8	59.43±3.9	61.20±2.4	55.37±3.5	55.57±2.7	48.27±2.1	56.11±2.8	53.78±3.4	59.04
MetaICL† (Min et al., 2022a)	88.80±5.0	64.22±2.0	62.39±3.4	57.34±1.9	59.18±2.8	50.46±2.5	57.90±1.8	57.13±2.4	62.18
SelfSup.† (Chen et al., 2022a)	88.55±3.0	64.24±2.0	63.42±2.2	55.70±1.8	58.93±2.6	48.08±2.5	58.01±2.5	58.28±3.2	61.90
KICT†	**92.18±2.9**	**71.32±0.7**	**71.23±1.0**	**62.89±1.2**	**66.10±4.2**	**58.33±3.8**	**64.90±5.5**	**69.27±5.7**	**69.53**

Table 12: The 8-shot performance (%) on GPT-2 (urge) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	SST-2 acc	MRPC f1	MNLI acc	QNLI acc	RTE acc	CB acc	TREC acc	AGNews acc	Avg.
Full Data									
Fine Tuning (RoBERTa-large)	95.00	91.40	89.80	93.30	80.90	90.50	97.40	94.70	91.63
Few-shot Labeled Data (8-shot)									
ICL (Brown et al., 2020)	79.43±7.2	56.72±2.3	59.28±2.4	55.37±3.2	56.01±5.0	44.48±1.8	54.10±4.1	47.95±6.6	56.67
CBU (Zhao et al., 2021)	83.77±4.4	65.38±3.9	58.49±2.8	55.88±3.9	56.26±2.8	53.89±1.7	56.37±3.7	58.20±2.8	61.03
KATE (Liu et al., 2022)	83.18±3.8	59.83±3.9	62.40±2.4	55.87±3.5	55.81±2.7	48.83±2.1	56.98±2.8	54.32±3.4	59.65
MetaICL† (Min et al., 2022a)	90.03±5.0	64.72±2.0	62.99±3.4	57.94±1.9	59.81±2.8	51.29±2.5	58.50±1.8	58.12±2.4	62.93
SelfSup.† (Chen et al., 2022a)	88.59±3.0	64.24±2.0	64.42±2.2	56.60±1.8	59.22±2.6	49.58±2.5	59.33±2.5	59.48±3.2	62.77
KICT†	**92.38±2.9**	**71.92±0.7**	**71.83±1.0**	**63.21±1.2**	**66.83±4.2**	**58.70±3.8**	**65.38±5.5**	**70.42±5.7**	**70.08**

Table 13: The 8-shot performance (%) on OPT (large) of different learning settings with standard deviations over text classification benchmarks. † denotes the method involves parameters update for ICL.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	23.70±3.7	49.20±1.9	43.10±3.4	37.30±3.0	38.34
CBU (Zhao et al., 2021)	26.37±3.1	52.90±2.8	46.88±2.0	41.38±2.9	41.89
KATE (Liu et al., 2022)	26.89±3.2	52.88±3.1	46.93±3.7	41.35±2.8	42.01
MetaICL† (Min et al., 2022a)	27.40±2.7	52.74±3.3	46.63±2.9	42.51±3.0	42.32
SelfSup.† (Chen et al., 2022a)	27.33±3.1	52.91±3.1	46.97±2.9	42.71±3.2	42.48
KICT†	**28.78±2.6**	**53.10±2.9**	**47.72±2.3**	**43.88±2.2**	**43.37**

Table 14: The 8-shot performance (%) on GPT-2 (small) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	29.15±2.4	55.78±3.1	49.12±3.1	42.11±2.7	44.04
CBU (Zhao et al., 2021)	31.58±3.9	57.01±2.6	51.28±2.8	45.70±4.4	46.39
KATE (Liu et al., 2022)	31.18±4.1	56.70±3.0	49.13±3.4	44.54±3.3	45.39
MetaICL† (Min et al., 2022a)	32.16±3.2	57.64±2.6	53.26±3.1	48.91±2.9	47.99
SelfSup.† (Chen et al., 2022a)	33.44±3.2	56.18±3.5	51.90±2.7	49.10±3.1	47.66
KICT†	**37.05±2.8**	**59.35±2.4**	**55.08±2.9**	**53.18±3.2**	**51.17**

Table 16: The 8-shot performance (%) on GPT-2 (urge) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	25.38±3.1	52.10±3.2	45.58±3.3	38.47±2.7	40.38
CBU (Zhao et al., 2021)	28.40±3.2	53.64±2.6	47.81±4.0	43.20±2.2	42.68
KATE (Liu et al., 2022)	28.38±3.1	54.26±3.3	46.70±3.7	41.98±4.1	42.83
MetaICL† (Min et al., 2022a)	29.67±2.9	54.37±2.5	48.79±2.4	45.11±3.1	44.49
SelfSup.† (Chen et al., 2022a)	29.36±3.0	54.10±2.2	48.47±2.7	44.06±3.1	44.00
KICT†	**34.81±3.0**	**56.38±2.9**	**51.18±2.8**	**46.00±3.5**	**47.09**

Table 15: The 8-shot performance (%) on GPT-2 (medium) of different learning settings with standard deviations over question answering benchmarks.

Baselines	ComQA acc	Quartz acc	SQuAD em	Quoref em	Avg.
Full Data					
Fine Tuning (RoBERTa-large)	72.10	76.90	86.50	78.70	78.55
Few Labeled Data (8-shot)					
ICL (Brown et al., 2020)	30.42±2.2	56.19±3.2	48.73±3.0	44.18±3.7	44.88
CBU (Zhao et al., 2021)	32.16±2.7	58.02±2.8	53.11±2.7	47.35±2.0	47.66
KATE (Liu et al., 2022)	33.32±3.6	58.90±2.9	50.65±2.4	46.12±3.5	47.25
MetaICL† (Min et al., 2022a)	33.96±3.4	58.64±2.4	54.11±2.4	48.12±2.7	48.71
SelfSup.† (Chen et al., 2022a)	34.42±3.0	58.12±3.0	54.92±2.7	49.53±1.8	49.25
KICT†	**39.22±2.8**	**61.71±2.4**	**59.67±2.1**	**54.40±3.1**	**53.75**

Table 17: The 8-shot performance (%) on OPT (large) of different learning settings with standard deviations over question answering benchmarks.