Valuing Training Data via Causal Inference for In-Context Learning

Xiaoling Zhou, Wei Ye, Zhemg Lee, Lei Zou, and Shikun Zhang

***Abstract*—In-context learning (ICL) empowers large pre-trained language models (PLMs) to predict outcomes for unseen inputs without parameter updates. However, the efficacy of ICL heavily relies on the choice of demonstration examples. Randomly selecting from the training set frequently leads to inconsistent performance. Addressing this challenge, this study takes a novel approach by focusing on training data valuation through causal inference. Specifically, we introduce the concept of average marginal effect (AME) to quantify the contribution of individual training samples to ICL performance, encompassing both its generalization and robustness. Drawing inspiration from multiple treatment effects and randomized experiments, we initially sample diverse training subsets to construct prompts and evaluate the ICL performance based on these prompts. Subsequently, we employ Elastic Net regression to collectively estimate the AME values for all training data, considering subset compositions and inference performance. Ultimately, we prioritize samples with the highest values to prompt the inference of the test data. Across various tasks and with seven PLMs ranging in size from 0.8B to 33B, our approach consistently achieves state-of-the-art performance. Particularly, it outperforms Vanilla ICL and the best-performing baseline by an average of 14.1% and 5.2% , respectively. Moreover, prioritizing the most valuable samples for prompting leads to a significant enhancement in performance stability and robustness across various learning scenarios. Impressively, the valuable samples exhibit transferability across diverse PLMs and generalize well to out-of-distribution tasks.**

***Index Terms*—In-context learning, data valuation, causal inference, average marginal effect, elastic net regression.**

I. Introduction

THE remarkable linguistic capabilities and extensive world knowledge embedded in large pre-trained language models (PLMs) [1], [2], [3], [4], [5] have recently promoted the emergence of a novel approach known as in-context learning (ICL), which represents a new paradigm in natural language understanding. In this paradigm, as depicted in Fig. 1, a PLM is presented with a prompt, typically comprising a few training examples, along with a test instance, and directly generates the output for the test instance without any parameter updates. As a new paradigm, ICL presents compelling advantages, facilitating natural language interaction with PLMs [6], [7], as well as reducing computational costs [8], [9].

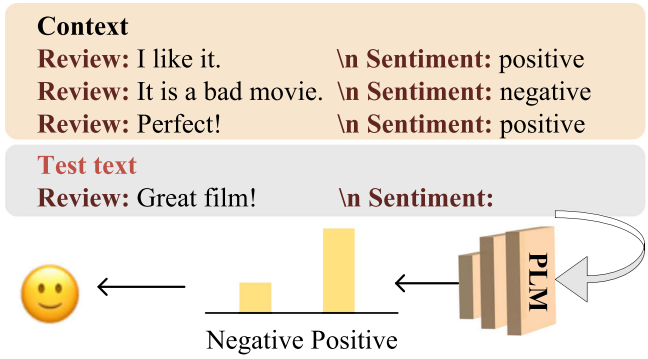

Fig. 1. Illustration of ICL for sentiment classification.

While ICL holds promise, its effectiveness hinges upon the quality of the provided demonstration examples. Randomly sampled in-context examples often display large instability and result in poor performance [6], [10]. Therefore, data curation [11], [12], [13], [14] plays a pivotal role in the ICL process, enabling the utilization of high-quality training samples as demonstrations, consequently leading to favorable outcomes. Numerous previous studies have focused on demonstration selection, encompassing metric-based methods (such as similarity and entropy) [10], [15], [16], training dense retrievers [9], [17], and active learning-based approaches [18], [19]. However, these methods can not effectively and accurately capture how each sample's presence influences ICL predictions, as well as overlooking correlations among different demonstration examples—a challenge widely acknowledged as NP-hard. Additionally, methods that rely on training dense extractors, as well as those employing active learning, often entail high computational costs for training and require additional annotations that may introduce labeling biases during the valuation process.

In this study, we tackle this challenge for the first time from the perspective of training data valuation through causal inference, considering the inclusion of each data source as a form of treatment. Specifically, quantifying the impact of a training point on ICL inference involves posing a counterfactual question: "*what would happen to the inference performance if this training point was excluded from the prompt?*" Answering this question

Received 22 July 2024; revised 15 January 2025; accepted 25 February 2025. Date of publication 28 February 2025; date of current version 1 May 2025. This work was supported by the National Key Research and Development Program of China under Grant 2023YFC3304404. Recommended for acceptance by Liqiang Nie. *(Corresponding authors: Wei Ye; Shikun Zhang.)*

Xiaoling Zhou, Wei Ye, and Shikun Zhang are with the National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China (e-mail: xiaolingzhou@stu.pku.edu.cn; wye@pku.edu.cn; zhangsk@pku.edu.cn).

Zhemg Lee is with Tianjin University, Tianjin 300072, China (e-mail: zhemglee@tju.edu.cn).

Lei Zou is with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China (e-mail: zoulei@pku.edu.cn).

Digital Object Identifier 10.1109/TKDE.2025.3546761

necessitates prompt modifications and utilizing these adjusted prompts for re-inference. Building on this novel perspective, we introduce the concept of average marginal effect (AME) [20] to evaluate the contribution of each training sample, which is defined as the expected marginal effect contributed by a data point to the model behavior over randomly selected subsets of the data and can be utilized to quantify the contribution of each training data point to the specific behavior of the trained model. Intuitively, a data point has a large AME when adding it to the training data impacts the behavior under study, regardless of the presence of other data points. This approach mitigates the issue of marginal contributions being near zero, which is commonly encountered in traditional methods [21]. Furthermore, by examining the marginal effects across subsets of varying sizes, it offers a more comprehensive representation of a data point's contribution under different conditions, ensuring that even subtle influences on the training set are effectively captured. To ensure a comprehensive evaluation of data values, we define two types of utilities concerning model generalization and robustness, respectively. According to our definition, a training sample with a high value signifies a significant positive influence on the generalization and robustness performance of PLMs. In contrast to previous demonstration selection methods, our approach directly quantifies the impact of each training sample on ICL performance, while also considering their combined effects, resulting in a more accurate and reasonable valuation of the training data. Furthermore, our approach eliminates the need for additional data annotations, thereby preventing the introduction of labeling bias.

To collectively calculate the AME values of all training data, we formulate their estimation as a specific linear regression problem, considering the composition of training subsets and the inference performance. However, computing the AME values exactly remains computationally expensive, as it requires conducting ICL inference with an infinite number of prompts constructed from various training subsets. Consequently, we employ Elastic Net regression, which is well-known for its efficacy in managing sparse solutions and ensuring parameter stability [22], [23], to tackle this regression problem. This approach notably improves computational efficiency by sampling fewer training subsets. Moreover, only a linear regression model needs to be trained, thereby improving training efficiency in comparison to previous training-based demonstration selection methods. Once the values of all training data, regarding both model generalization and robustness performance, are obtained, we select those with the highest combined values to construct the task-specific prompt for inferring the test data.

Extensive experiments have been conducted across various classification tasks and with seven PLMs ranging in size from 0.8B to 33B, demonstrating that our proposed AME-ICL[1] approach significantly outperforms previous prompt retrieval approaches in terms of both effectiveness and efficiency. On average, AME-ICL surpasses Vanilla ICL, which randomly samples demonstrations from the training data, by 14.1%. Moreover, it exceeds the best performance of comparative baselines by 5.2%. Additionally, a significant reduction in performance variance is observed across various learning scenarios, underscoring AME-ICL's capability to enhance the stability of ICL performance. Notably, the calculated data values have been verified to be transferable across different PLMs and generalize well to out-of-distribution (OOD) tasks.

Overall, our main contributions are summarized as follows:

- We conduct a pioneer exploration by introducing AME to quantify the contribution of each training sample to ICL inference. The AME value is defined as the expected marginal effect on ICL performance, considering both the generalization and robustness performance of PLMs. This approach, grounded in causal inference, adeptly accounts for the interdependencies among various training samples, thus yielding a precise measurement of data values.
- We establish the AME-ICL framework to calculate the values of all training data and curate the prompt with the most valuable samples for ICL inference. Within our framework, the computation of all AME values is formulated as a linear regression problem, which is addressed using Elastic Net. Our framework is straightforward, effective, and scalable, enabling seamless integration with various PLMs.
- We conduct extensive experiments on both classification and commonsense reasoning tasks across ten large PLMs, demonstrating that AME-ICL consistently achieves state-of-the-art (SOTA) performance in terms of both generalization and robustness. Moreover, it enhances prediction stability across various scenarios, encompassing imbalanced labels in prompts, OOD tasks, and variations in the number, template, and permutation of demonstrations.

II. Related Work

A. In-Context Learning

Brown et al. [5] showcased the ability of PLMs in ICL, wherein predictions are formulated solely based on a concatenation of training instances for few-shot learning, without parameter updates. Building upon this foundation, subsequent studies [24], [25], [26] have extended and refined this approach, resulting in promising outcomes across a spectrum of tasks. For example, Wies et al. [24] proposed a PAC-based framework for in-context learnability and demonstrated that when the pretraining distribution consists of a mixture of latent tasks, these tasks can be effectively learned through ICL. Moreover, Qin et al. [27] proposed an iterative demonstration selection method, which progressively selects examples that are both diverse and strongly correlated with the test sample to serve as demonstrations for ICL. Additionally, Jiang et al. [28] calibrated the in-context predictive distribution by adjusting the label marginal, which is estimated via Monte-Carlo sampling over the in-context model. Recently, significant progress has also been made in the understanding of ICL. For example, Zhang et al. [29] observed that with an increasing number of ICL examples, models initially exhibit increased miscalibration before achieving better calibration and miscalibration tends to arise in low-shot settings. Yan et al. [30] explored the elusive mechanism underpinning ICL and revealed a principle that strengthens the relationship

[1]Our code is available at https://github.com/xiaolingzhou98/AME-ICL.

between two tokens based on their contextual co-occurrences. Min et al. [31] demonstrated that the model does not heavily rely on the ground truth input-label mapping provided in the demonstrations.

The ICL approach has gained widespread popularity across various applications. For instance, Qu et al. [32] successfully leveraged ICL to generate coarse-grained layouts conditioned on a given textual prompt using large PLMs. Similarly, Wei et al. [7] employed ICL to enhance performance across a diverse set of tasks, including arithmetic, commonsense reasoning, and symbolic reasoning. However, a primary issue for the ICL approach is its inconsistent performance, which is sensitive to various factors. For instance, models have exhibited a tendency to excessively rely on either the most frequent labels (majority bias) [33] or labels appearing later in a prompt (recency bias) [34]. Moreover, a large performance gap between using the most negative in-context examples and the most positive ones has been clearly revealed [35], demonstrating that the selection of demonstrations significantly influences performance. Furthermore, research [36] has revealed that large PLMs can override semantic priors when presented with in-context exemplars that contradict priors, despite the stronger semantic priors that larger models may hold. Additional research has also revealed that the accuracy of input-label mapping has minimal impact [37], while the diversity of examples is of greater importance [19]. Nevertheless, selecting the most effective samples for prompting and ordering them appropriately is crucial for enhancing the performance and stability of ICL.

B. Prompt Retrieval

The efficacy of ICL heavily hinges on the selected demonstration examples. Previous research on ICL has predominantly concentrated on retrieving demonstration examples at the instance level. For instance, Qin et al. [27] iteratively selected examples that are diverse but still strongly correlated with the test sample as ICL demonstrations. Rubin et al. [9] and Shi et al. [38] trained the prompt retriever based on feedback from PLMs for semantic parsing. Moreover, Li et al. [39] cast various tasks' training signals into a unified list-wise ranking formulation by PLM's feedback and proposed a multi-task list-wise ranking training framework to train a unified demonstration retriever. Furthermore, Levy et al. [40] posited that diverse demonstrations would benefit ICL inference. Additionally, Agrawal et al. [41] demonstrated that both the translation quality and the domain of in-context examples are crucial for machine translation tasks. They thus proposed an approach that incorporates similar examples based on n-gram overlap with the test source.

Another line to retrieve prompts involves active learning [42]. For example, Zhang et al. [18] approached demonstration selection for ICL by framing it as a sequential decision problem. They proposed a reinforcement learning algorithm aimed at identifying generalizable policies for selecting demonstration examples. Moreover, Margatina et al. [43] addressed the issue of identifying the most informative demonstrations for few-shot learning by approaching it as a pool-based active learning problem over a single iteration. However, these methods consume significant computational resources and are sensitive to noise. This study explores a new path for prompt retrieval. Specifically, we propose a novel method for estimating the AME value of each training sample to evaluate its contribution to the generalization and robustness of ICL predictions. Based on these estimates, we then select the most valuable samples to construct prompts.

Compared to previous approaches for demonstration selection, our method offers several notable advantages:

- In contrast to approaches focusing on selecting instance-specific demonstration samples, this study underscores task-level example selection, with the aim of identifying valuable examples that broadly and effectively represent the task. Consequently, the prediction performance for the entire task can be enhanced with these selected samples.
- Our method leverages the concept of AME, which quantifies the expected marginal impact of each training sample on ICL performance, to directly assess the contribution of each training sample while accounting for the correlations among individual demonstration examples. This approach leads to a more accurate and meaningful valuation of the training data.
- Unlike methods that rely on training dense extractors and active learning techniques, our approach involves only solving a sparse linear regression, eliminating the need for additional label annotations, which is more efficient and straightforward.

C. Data Valuation

The objective of data valuation is to assess the individual contribution of each data point to model behavior [14], [44]. This study assesses the value of each training example within the ICL process. Current data valuation methodologies can be categorized into four main folds: marginal contribution-based methods [45], [46], gradient-based methods [21], [47], importance weight-based methods [48], and out-of-bag (OOB) estimation-based methods [49]. Among these, marginal contribution-based methods assess data values by measuring the difference in utility with and without each data point under consideration. A larger difference indicates a higher value. Notable methods include leave-one-out [50], Data Banzhaf [51], and a range of Shapley value-based approaches [52], [53] such as Data Shapley [46], Beta Shapley [45], and AME [20], [54], [55]. Notably, this type of method typically entails training different models on extensive training subsets. Nevertheless, AME significantly improves computational efficiency compared with others by utilizing a sparse linear estimator to calculate the values associated with training data, thereby enhancing its potential for practical applications. Gradient-based methods evaluate data value by analyzing the change in utility when the weight of the data point is adjusted. Prominent methods here include the influence function [21], datamodels [56], and LAVA [47]. However, this kind of approach may be affected by the noise of gradient estimation. Moreover, importance weight-based methods assign a weight to each data point during training, with the weight serving as its value. These methods are specially tailored for machine learning applications with high computational complexity. DVRL [48]

is a notable example, utilizing reinforcement learning to learn weights. OOB estimation-based methods are also specifically devised for machine learning tasks, which may be affected by sample selection bias. A key method is Data-OOB [49], which computes data point contribution using out-of-bag accuracy.

Our method falls within the category of marginal contribution-based approaches considering their direct measurement and effectiveness. Specifically, we leverage the AME [20] concept to assess the contribution of each training sample. However, in contrast to previous AME methods [20], [54], [55], our proposed approach introduces the following novel extensions:

- For achieving a more comprehensive training data valuation, we propose two utility measures that correspond to the model generalization and robustness performance of PLM predictions, respectively. The data values associated with these two aspects are weighted and summed to select the most significant training data for prompt construction.
- Our method innovatively focuses on the ICL procedure, where training examples act as prompts without updating model parameters. As a result, the need to train multiple models on diverse training subsets is eliminated.
- To enhance stability and alleviate aggressive coefficient shrinkage in LASSO, we estimate the AME values for training data using Elastic Net regression which involves both L_1 and L_2 regularizers, selected for its ability to handle sparse solutions and ensure parameter stability.

III. Methodology

A. In-Context Learning With PLMs

Assuming the existence of a training set $\mathcal{D}^{tr}$, a validation set $\mathcal{D}^{dev}$, and a held-out test set $\mathcal{D}^{te}$, our objective is to identify the most valuable training samples from $\mathcal{D}^{tr}$ based on the prediction performance observed on $\mathcal{D}^{dev}$. These identified valuable samples then serve as prompts for the inference of PLMs. Consequently, the inference performance on $\mathcal{D}^{te}$ can be enhanced by leveraging these valuable samples as prompts.

Following previous ICL research [33], [57], [58], [59], this study focuses on the classification task. Specifically, considering a PLM G, given an input text $\boldsymbol{x}$ and a candidate answer set $L=\{y_1, y_2, \ldots, y_{|L|}\}$ with $|L|$ classes, we aim to predict the answer $\hat{y}$ for $\boldsymbol{x}$ based on $\mathcal{M}$ selected valuable training examples: $\mathcal{C}=\{e_1, e_2, \ldots, e_{\mathcal{M}}\}$, where each e_i represents a training example $(\boldsymbol{x}_i^{tr}, y_i^{tr})$ and $\mathcal{M}$ denotes the number of demonstration examples. Formally, give a model G, we first compute the probability of each answer y_j:

$$P_G\left(y_j \mid \mathcal{C}, \boldsymbol{x}\right). \tag{1}$$

Subsequently, the ultimate prediction $\hat{y}$, characterized by the highest probability is chosen from the candidate answer set L:

$$\hat{y} = \arg\max_{y_j \in L} P_G\left(y_j \mid \mathcal{C}, \boldsymbol{x}\right). \tag{2}$$

The prediction accuracy for the test set Acc^{te} is utilized to evaluate the performance, which is calculated as

$$Acc^{te} = \frac{1}{|\mathcal{D}^{te}|} \sum_{i=1}^{|\mathcal{D}^{te}|} \mathbb{I}(\hat{y}_i = y_i), \tag{3}$$

where $|\mathcal{D}^{te}|$ denotes the size of the test set $\mathcal{D}^{te}$ and $\mathbb{I}(\cdot)$ is a indicator function.

During the ICL process, we explore two settings following those outlined in [10]: one where the training samples are labeled, and another where they are unlabeled. The first setting assumes access to a labeled training dataset, denoted as $\mathcal{D}_{\mathcal{L}}^{tr} = \{(\boldsymbol{x}_i^{tr}, y_i^{tr})\}_{i=1}^{N}$, along with a smaller labeled validation set $\mathcal{D}^{dev}$. The second setting is closer to the true few-shot learning setup [60], where we only have a labeled validation set $\mathcal{D}^{dev}$ and an unlabeled training set $\mathcal{D}^{tr} = \{\boldsymbol{x}_i^{tr}\}_{i=1}^{N}$. In this setup, each input $\boldsymbol{x}_i^{tr}$ is paired with a randomly sampled label $\tilde{y}_i^{tr} \in L$ to create the training set $\mathcal{D}_{\mathcal{U}}^{tr} = \{(\boldsymbol{x}_i^{tr}, \tilde{y}_i^{tr})\}_{i=1}^{N}$. In both scenarios, our objective is to select the most valuable samples from either $\mathcal{D}_{\mathcal{L}}^{tr}$ or $\mathcal{D}_{\mathcal{U}}^{tr}$ to construct prompts for ICL inference. Additionally, the labeled test set $\mathcal{D}^{te}$ is used to evaluate the effectiveness of our approach.

B. Training Data Valuation

1) Utility Definition: Our objective is to explore the impact of each training point on the performance of ICL inference. Therefore, we define $Q_G(\mathcal{S})$ as the utility of a specific behavior exhibited by PLM G when utilizing samples from training subset $\mathcal{S}$ as demonstrations. We consider two definitions for the utility $Q_G(\mathcal{S})$, each pertaining to model generalization and robustness respectively, thereby ensuring a comprehensive evaluation of the contribution of each training sample.

First, we define the utility Q_G as the prediction accuracy achieved on the validation set $\mathcal{D}^{dev}$. To mitigate the effect of the permutation of demonstration samples on the ICL performance, we consider a total of $\mathcal{O}$ permutations for the samples in each subset $\mathcal{S}$. Let $\mathcal{S}^o$ denote the prompt constructed using samples in $\mathcal{S}$ with a specific order o. The generalization utility Q_G^g is calculated as follows:

$$Q_G^g(\mathcal{S}) = \frac{1}{\mathcal{O}|\mathcal{D}^{dev}|} \sum_{o=1}^{\mathcal{O}} \sum_{i=1}^{|\mathcal{D}^{dev}|} \mathbb{I}(\hat{y}_i(\mathcal{S}^o) = y_i), \tag{4}$$

where $|\mathcal{D}^{dev}|$ represents the size of $\mathcal{D}^{dev}$ and $\hat{y}_i(\mathcal{S}^o)$ denotes the predicted label of the ith sample in $\mathcal{D}^{dev}$ when employing the prompt $\mathcal{S}^o$ for inference:

$$\hat{y}_i(\mathcal{S}^o) = \arg\max_{y_j \in L} P_G(y_j | \boldsymbol{h}_{\tilde{\boldsymbol{x}}_i^o}), \tag{5}$$

where $\boldsymbol{h}_{\tilde{\boldsymbol{x}}_i^o}$ represents the hidden state of the last block at the final position for the contextual input $\tilde{\boldsymbol{x}}_i^o = [\mathcal{S}^o, \boldsymbol{x}_i]$. Due to the input length limitations for PLMs, if the size of $\mathcal{S}$ is too large, it will be truncated in applications.

Moreover, we define the utility Q_G as the robust accuracy on the validation set, which is computed as follows:

$$Q_G^r(\mathcal{S}) = \frac{1}{\mathcal{O}|\mathcal{D}^{dev}|}\sum_{o=1}^{\mathcal{O}}\sum_{i=1}^{|\mathcal{D}^{dev}|}\mathbb{I}(y_i^r(\mathcal{S}^o) = y_i), \tag{6}$$

where $y_i^r(\mathcal{S}^o)$ represents the prediction for the perturbed feature of the ith sample in $\mathcal{D}^{dev}$ using prompt $\mathcal{S}^o$ for inference:

$$y_i^r(\mathcal{S}^o) = \arg\max_{y_j \in L} P_G(y_j|\tilde{\boldsymbol{h}}_{\tilde{\boldsymbol{x}}_i^o}), \tag{7}$$

where $\tilde{\boldsymbol{h}}_{\tilde{\boldsymbol{x}}_i^o}$ is the perturbed feature of $\tilde{\boldsymbol{x}}_i^o$, calculated using

$$\tilde{\boldsymbol{h}}_{\tilde{\boldsymbol{x}}_i^o} = \boldsymbol{h}_{\tilde{\boldsymbol{x}}_i^o} + \epsilon\frac{\partial \ell_i}{\partial \boldsymbol{h}_{\tilde{\boldsymbol{x}}_i^o}}. \tag{8}$$

Here, ℓ_i represents the Cross-Entropy loss of the ith sample in $\mathcal{D}^{dev}$ and ϵ denotes the perturbation bound. This utility definition ensures an assessment of the contribution of each training sample to the robustness performance of PLMs.

Consequently, when describing our technique, we will need to calculate the generalization and robustness utilities (i.e., $Q_G^g(\mathcal{S})$ and $Q_G^r(\mathcal{S})$) on various training subsets, where each represents the utility result when applied to a PLM G with samples from subset $\mathcal{S}$ serving as prompts.

2) AME Estimation: Having established the utility, we now evaluate the contribution of each training point $e_i = (\boldsymbol{x}_i^{tr}, y_i^{tr})$ to the corresponding generalization and robustness utilities. According to the counterfactual question "*what would happen to the inference performance if the training point e_i was excluded from the prompt?*", we measure this change by computing $Q_G(\mathcal{S}) - Q_G(\mathcal{S}\backslash\{e_i\})$, signifying the change in utility when the data point e_i is included and excluded from the prompts. Drawing inspiration from the assessment of multiple treatment effects in the causal inference literature [61], we compute the average marginal contributions of including each data point e_i across training subsets of varying sizes. Consequently, the AME value of e_i (V_i) is defined as its expected marginal effect [54] on subsets sampled from the training distribution $\mathcal{N}^{tr}$:

$$V_i = \mathbb{E}_{\mathcal{S}^{e_i}\sim\mathcal{N}^{tr}}[Q_G(\mathcal{S}^{e_i} + \{e_i\}) - Q_G(\mathcal{S}^{e_i})], \tag{9}$$

where $\mathcal{S}^{e_i}$ represents a subset of training data excluding e_i, drawn from $\mathcal{N}^{tr}$. The choice of the sampling distribution $\mathcal{N}^{tr}$ plays a crucial role in determining both the specific aspect that AME measures and the efficiency with which it can be estimated. Consequently, we construct subsets from training data by assigning each data point (excluding the one being measured, e_i) a sampling probability p drawn from a distribution $\mathcal{P}$[2].

To calculate AME values more efficiently, we adopt the assumption of sparsity, which posits that the number of data points with non-zero AMEs is relatively small in comparison to the total number of instances in the training data. This assumption is suitable for our scenario as only a limited number of samples can be selected as demonstrations due to the input length limitations of PLMs. Then, we reframe the estimation of all $\boldsymbol{V}$ values as a specific linear regression problem [54], [55], [62]. Inspired by randomized experiments, we initiate by generating K subsets of the training data, denoted as $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K$. Each subset $\mathcal{S}_k$ is sampled by first selecting a probability p drawn from the distribution $\mathcal{P}$, then including each training data point with probability p. In our linear regression, the observation matrix $\boldsymbol{X}$ is a $K \times N$ matrix, i.e., $\boldsymbol{X} \in \mathbb{R}^{K\times N}$, where each row $\boldsymbol{X}[k,:]$ comprises N dimensions, one for each training data point, indicating its presence or absence in the sampled subset $\mathcal{S}_k$. Moreover, when constructing $\boldsymbol{X}$, it's essential to consider the sampling probability p. Therefore, we adjust the features based on p to counterbalance the variance weighting. Specifically, for $r \sim \text{Bernoulli}(p)$, we set $\boldsymbol{X}[k,i] = \frac{r}{p} - \frac{1-r}{1-p}$. Additionally, the response vector $\boldsymbol{Y}$ is of size K, i.e., $\boldsymbol{Y} \in \mathbb{R}^K$, where each element $\boldsymbol{Y}[k]$ represents the utility score measured for the sampled subset $\mathcal{S}_k$, i.e., $\boldsymbol{Y}[k] = Q_G(\mathcal{S}_k)$. Consequently, our linear regression problem is constructed as follows:

$$\boldsymbol{V}^* = \arg\min_{\boldsymbol{V}\in\mathbb{R}^N} \mathbb{E}[(\boldsymbol{Y} - <\boldsymbol{V}, \boldsymbol{X}>)^2], \tag{10}$$

where $\boldsymbol{V}^* \in \mathbb{R}^N$ represents the optimal linear fit on the $(\boldsymbol{X}, \boldsymbol{Y})$ dataset, which contains the estimated AME values of all training points.

It is anticipated that a reduced number of subsets (smaller than the total number of training samples N) will be sampled to decrease the inference times of PLMs across various prompts. However, this approach may result in an under-determined regression problem, as the number of equations is fewer than the number of variables [63], [64]. Consequently, we leverage

Algorithm 1: Data Valuation.

Input: Training data $\mathcal{D}^{tr} = \{e_i\}_{i=1}^N$, validation data $\mathcal{D}^{dev}$, PLM G, subset count K, order count $\mathcal{O}$, probability distribution $\mathcal{P}$, utility Q_G, α, β, and others.

Output: Values of all training data.

Initialize $\boldsymbol{X} \leftarrow \text{zeros}(K, N)$ and $\boldsymbol{Y} \leftarrow \text{zeros}(K)$;
for $k \leftarrow 1$ *to* K **do**
  Initialize $\mathcal{S}_k$ as an empty set;
  Sample p from $\mathcal{P}$;
  for $i \leftarrow 1$ *to* N **do**
    $r \sim \text{Bernoulli}(p)$;
    if $r = 1$ **then**
      $\mathcal{S}_k \leftarrow \mathcal{S}_k + \{e_i\}$
    $\boldsymbol{X}[k,i] \leftarrow \frac{r}{p} - \frac{1-r}{1-p}$;
  for $o \leftarrow 1$ *to* $\mathcal{O}$ **do**
    Construct prompt $\mathcal{S}_k^o$ using samples in subset $\mathcal{S}_k$ sorted in order o;
    Inference on $\mathcal{D}^{dev}$ using PLM G with prompt $\mathcal{S}_k^o$;
  Calculate the utility $Q_G(\mathcal{S}_k)$ using Eq. (4) or (6);
  $\boldsymbol{Y}[k] \leftarrow Q_G(\mathcal{S}_k)$;
$\boldsymbol{V}_{EN} = \arg\min_{\boldsymbol{V}\in\mathbb{R}^N} \left((\boldsymbol{Y} - <\boldsymbol{V}, \boldsymbol{X}>)^2 + \alpha||\boldsymbol{V}||_1 + \beta||\boldsymbol{V}||_2\right)$

[2]Considering the input length limitations of PLMs, we utilize a uniform distribution with small probabilities $\mathcal{P} = \text{Uniform}\{0.1, 0.2, 0.3, 0.4\}$ in our experiments.

Algorithm 2: AME-ICL.

Input: Test data $\mathcal{D}^{te}$, training data $\mathcal{D}^{tr}$, demonstration count $\mathcal{M}$, PLM G, batch size $\mathcal{B}$, λ_V, and others.

Output: Inference accuracy on $\mathcal{D}^{te}$

Calculate two types of values $\boldsymbol{V}^g$ and $\boldsymbol{V}^r$ for each training sample using Algorithm 1;

Select top-$\mathcal{M}/|L|$ samples with the highest $\hat{\boldsymbol{V}}$ (calculated in Eq. (12)) values from $\mathcal{D}^{tr}$ for each class;

Sort the valuable samples in ascending order of $\hat{\boldsymbol{V}}$ to construct the prompt $\mathcal{C}=\{e_1, e_2, \cdots, e_{\mathcal{M}}\}$;

for $i \leftarrow 1$ *to* $\lfloor \frac{|\mathcal{D}^{te}|}{\mathcal{B}} \rfloor$ **do**

  Sample $\mathcal{D}_i^{te}$ containing $\mathcal{B}$ instances from $\mathcal{D}^{te}$;

  Inference on $\mathcal{D}_i^{te}$ using G with prompt $\mathcal{C}$;

  $\mathcal{A}_i \leftarrow \sum_{j=1}^{\mathcal{B}} \mathbb{I}(\hat{y}_j(\mathcal{C}) = y_j)$;

$Acc^{te} \leftarrow \frac{1}{\mathcal{B}\times\lfloor \frac{|\mathcal{D}^{te}|}{\mathcal{B}} \rfloor} \sum_{i=1}^{\lfloor \frac{|\mathcal{D}^{te}|}{\mathcal{B}} \rfloor} \mathcal{A}_i$

sparsity by integrating the L_1 norm regularization term into this regression problem. Moreover, recognizing the strong correlation among different training samples, we further introduce an L_2 norm regularization term to enhance parameter stability and model resilience against noise. Consequently, our linear regression problem can be transformed into the following Elastic Net regression:

$$\boldsymbol{V}_{EN} = \arg\min_{\boldsymbol{V}\in\mathbb{R}^N} \left((\boldsymbol{Y} - < \boldsymbol{V}, \boldsymbol{X} >)^2 + \alpha||\boldsymbol{V}||_1 + \beta||\boldsymbol{V}||_2 \right). \tag{11}$$

The parameter α and β controls the strengths of L_1 and L_2 regularization terms, respectively. The algorithm for our data valuation process is outlined in Algorithm 1. Consequently, by employing Elastic Net regression twice with different values of $\boldsymbol{Y}$ (i.e., Q_G^g and Q_G^r), each training sample is associated with two values, which indicate the sample's contribution to enhancing the generalization and robustness of PLM predictions, respectively.

C. Prompt Construction

Once the values of training samples regarding the generalization and robustness performance are calculated, each training point e_i will be assigned two values: V_i^g and V_i^r. Here, V_i^g and V_i^r represent the values calculated using Q_G^g and Q_G^r, respectively. Then, for each training sample, its total value is computed as:

$$\hat{V}_i = V_i^g + \lambda_V V_i^r, \tag{12}$$

where λ_V is a hyperparameter, its value adjustable based on specific needs for generalization and robustness. Typically, λ_V can be set to 1, reflecting an equal emphasis on generalization and robustness.

Subsequently, samples with the highest $\hat{\boldsymbol{V}}$ values are prioritized to construct the prompt for inference. Thus, we opt to select the top-$\mathcal{M}/|L|$ training examples from each class, where $|L|$ represents the number of classes, and $\mathcal{M}$ denotes the number of demonstration examples in the prompt. This approach ensures a balanced class distribution within the prompt. Furthermore, considering insights from previous research [34], [65] that samples closer to the query carry greater importance, we arrange the samples in ascending order of their values $\hat{\boldsymbol{V}}$. This arrangement ensures that samples closest to the query are prioritized as the most valuable ones. Finally, the constructed prompt $\mathcal{C}$ is utilized during the inference phase on the test set $\mathcal{D}^{te}$. The pipeline of AME-ICL is depicted in Fig. 2, and the algorithm for AME-ICL is presented in Algorithm 2.

IV. Experimental Configuration

Our experimental investigation can be divided into three main components. In the first part, we compare AME-ICL with previously advanced demonstration selection methods to validate its capability to improve ICL performance. The second component comprises a series of analytical experiments aimed at validating the effectiveness of AME-ICL in various learning scenarios. In the third section, we delve into the efficiency of AME-ICL, as well as conducting ablation and sensitivity studies to gain deeper insights into each of its components.

A. Datasets and Models

Ten large PLMs, ranging in size from 0.8B to 33B, are employed to showcase the adaptability of AME-ICL across different model sizes. The involved PLMs in our study include GPTJ-6B [66], OPT-13B [67], GPT-2-0.8B, GPT-2-1.5B [68], GPT-Neo-2.7B [69], OPT-6.7B [67], LLaMA-33B [70], LLaMA-2-7B [71], LLaMA-2-13B, and LLaMA-3-8B [72].

Following previous research [10], [57], [58], our experiments are conducted on five classification tasks, including a sentiment analysis dataset, SST-2 [73], two natural language inference datasets, BoolQ [74] and Scicite [75], a subjectivity classification dataset, Subj [76], and a news classification dataset, AGNews [77]. Table I presents examples and label mappings for all five datasets. For each task, we utilize class-balanced $\mathcal{D}^{tr}$, $\mathcal{D}^{dev}$, and $\mathcal{D}^{te}$. We set $|\mathcal{D}^{tr}| = 1,000$ to ensure a diverse range of training examples for subset selection, and $|\mathcal{D}^{te}| = 1,000$ to facilitate reliable evaluation. $\mathcal{D}^{dev}$ comprises 50 examples per class. All three datasets are randomly sampled from the original training set and are mutually exclusive. Besides the above tasks, we also involve assessing the performance of our approach on two commonsense reasoning tasks, including CommonSenseQA [78] (CSQA) and OpenBookQA [79] (OBQA).

B. Compared Baselines

Previous studies [6], [27] have demonstrated that both the selection of demonstrations and their ordering significantly affect ICL performance and have proposed various advanced methods to enhance ICL performance by optimizing demonstration selection and ordering strategies [10], [18], [80]. Therefore, besides Vanilla ICL, we compare AME-ICL with seven demonstration selection and two demonstration permutation methods. The compared methods are described as follows. First, following

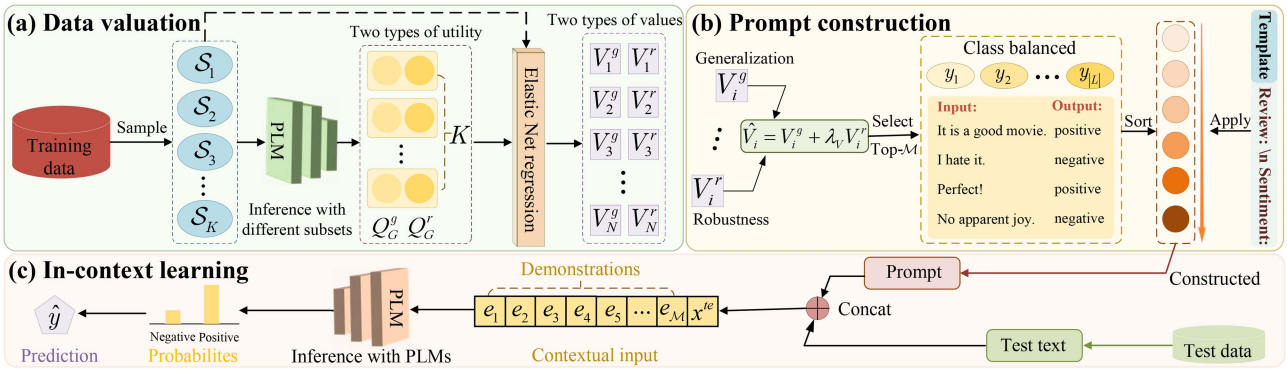

Fig. 2. Pipeline of AME-ICL consists of three main steps: training data valuation, prompt construction, and ICL inference. First, we compute the values of all training data in terms of both the generalization and robustness performance of ICL inference. Subsequently, we select the most valuable samples to construct a task-specific prompt. Finally, we employ the constructed prompt to infer test data. By utilizing the most valuable training samples as demonstrations, both the generalization and robustness performance of ICL predictions can be enhanced.

TABLE I
TEMPLATES AND LABEL MAPPINGS ACROSS DIFFERENT TASKS

Task	Example	Label mapping
SST-2	Review: contains no wit, only labored gags. Sentiment: negative	negative/positive
BoolQ	Exercise: read the text and answer the question by yes or no. Good Samaritan laws offer legal protection to people who give reasonable assistance... Question: do good samaritan laws protect those who help at an accident? yes	no/yes
Subj	Input: the tucks have a secret, they're immortal. Type: objective	objective/subjective
Scicite	Is the following citation from a scientific paper describing a method, a result, or background? However, how frataxin interacts with the Fe-S cluster biosynthesis components... Answer: background	method/result/background
AGNews	Article: Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers... Answer: business	world/sports/business/technology

To streamline the process, we ensure that all label words we employ comprise a single token, facilitating the straightforward calculation of the probability associated with each label.

Chang and Jia [10], we consider three demonstration selection methods based on ICL accuracy. ONESHOT initially conducting ICL with $\mathcal{M} = 1$, utilizing each training example individually as a prompt. Subsequently, the example's effectiveness is evaluated based on its corresponding ICL accuracy on $\mathcal{D}^{dev}$. This evaluation scheme aims to assess the extrapolation of ICL performance from $\mathcal{M} = 1$ to $\mathcal{M} > 1$. TOPPROMPTS-5 and TOPPROMPTS-10 aggregate examples from the top-5,10 prompts with the highest accuracy on the validation set. CONDACC [10] scores a training example based on its average ICL accuracy on the validation set when combined with random training examples. Another considered demonstration selection approach is DATAMODELS [10], which trains a datamodel to approximate an LLM's outcome for each sample in $\mathcal{D}^{dev}$. Besides, we compare AME-ICL with three metric-based selection approaches: K-Center Greedy [81], which assumes that training samples close in feature space have similar properties, thereby selecting samples with high similarities; GraNd [82], which selects the most informative examples based on the gradient norm expectations of samples; and Max-Entropy [18], which greedily selects examples to maximize classification entropy. Additionally, we compare AME-ICL with two demonstration ordering methods: PDO [80], which orders demonstrations based on the model's probability predictions, and GlobalE [6], which selects the ordering that minimizes the KL-divergence between the uniform distribution and the predicted label distribution. Furthermore, two other manners are also involved in comparison to further demonstrate the efficacy of our approach. RANDOM randomly selects a balanced training subset consisting of twenty examples and then chooses demonstration examples from this subset. Calibration (CALIB) [34] mitigates biases towards specific labels in PLMs. Finally, we extend AME-ICL and other compared methods that do not require labels to the unlabeled setup, denoted by the prefix UN.

C. Setups

During ICL inference, the batch size is set to 16, and the sequence length is configured to 256. For binary classification tasks, we set $\mathcal{M} = 4$ with balanced class distribution. For multiclass tasks (i.e., Scicite and AGNews), a training example per class is sampled to form the prompt. Our ablation studies also investigate the utilization of different numbers of demonstrations, namely $\mathcal{M} = \{1, 4, 8, 12\}$. The demonstration samples

are arranged in ascending order of their total values $\hat{V}$. But we also explore other permutations in the analytical experiments. For each task, a specific template is utilized for inference. Additionally, we examine the impact of different templates on the performance of AME-ICL following those outlined by Zhao et al. [34], which are listed in Table VIII. Each experiment is repeated using five random seeds. During data valuation, the number of subsets K is set to 100; ten random permutations are considered for each subset. ElasticNetCV is used, with the $l1$_ratio set to 0.5 which controls the relative contribution of the L_1 and L_2 regularization terms, and α set to 0.01 to control the strength of both regularization terms. As for the hyperparameters in AME-ICL, the perturbation bound ϵ is selected from the set $\{0.1, 0.2, 0.3\}$, and the parameter λ_V is chosen from the set $\{0.5, 1.0, 1.5\}$. We also conduct a sensitivity analysis of these two hyperparameters in our analytical experiments.

Recall that our objective is to select the most valuable training samples to create the prompt that enhances the generalization and robustness of PLMs. To ensure a fair comparison, we consider two settings in our experimental investigation. Initially, following the approach of Chang and Jia [10], we select the most valuable $\lceil 20/|L| \rceil$ samples from each class to create a stable set. Subsequently, we randomly sample 50 prompts from this selected subset and apply ICL on the test set $\mathcal{D}^{te}$. In this scenario, we report the average accuracy, standard deviation, and worst accuracy to ensure a comprehensive evaluation. In the second setting, we directly select the top-$\mathcal{M}$ samples to construct the prompts for inferring the test data $\mathcal{D}^{te}$. Both the average accuracy and standard deviation are reported.

V. Experimental Findings

A. Main Comparison Results

Table II presents the test set accuracy achieved using different demonstration selection approaches. AME-ICL consistently demonstrates the highest average accuracy with low standard variance, highlighting its ability to enhance the generalization capability and performance stability of PLMs. Overall, AME-ICL exhibits a 13.7% improvement over Vanilla ICL, which randomly selects demonstration examples from training data, on average. Compared to the best performance of the other approaches, AME-ICL outperforms them by 5.8%. Notably, our proposed AME-ICL consistently surpasses the calibration method, CALIB, underscoring the significance of selecting valuable samples as demonstrations to enhance ICL performance. Among the compared baselines, Vanilla ICL and RANDOM exhibit similarly lower performance. Moreover, ONESHOT outperforms Vanilla ICL and RANDOM on SST-2 and BoolQ, and performs comparably on other tasks, suggesting that selecting high-quality demonstrations is more important than simply increasing the number of demonstration examples. While applying prediction calibration enhances the average accuracy on certain tasks, it is not universally beneficial, particularly on Scicite. Methods like K-Center Greedy, GraNd, and Max-Entropy focus solely on one aspect, be it similarity, entropy, or gradient norm, when selecting demonstration samples. Their narrow focus generally hinders them from consistently attaining favorable outcomes. Additionally, CONDACC and DATAMODELS emerge as strong baselines for demonstration selection, while PDO serves as a strong baseline for demonstration ordering. Nevertheless, AME-ICL notably outperforms these approaches, suggesting that valuing training data based on the AME concept is more effective than that based on average accuracy and gradient.

AME-ICL consistently achieves the highest worst accuracy across various tasks, showcasing its effectiveness in improving the stability of ICL predictions. From the results in Table II, AME-ICL's worst accuracy exceeds that of the best-performing method among the baselines by 8.0% on average. Furthermore, compared to Vanilla ICL, AME-ICL exhibits an improvement of 21.0%. These findings suggest that the samples selected by our approach are more effective in enhancing ICL performance. Among the baselines, Vanilla ICL and RANDOM demonstrate comparable levels of instability, highlighting the pivotal role of demonstration selection in fortifying the stability of ICL predictions. The incorporation of calibration (CALIB) generally improves the worst accuracy across various tasks, emphasizing the role of prediction calibration in enhancing performance stability. It is worth noting that CONDACC and DATAMODELS are the most robust baselines for demonstration selection, while GlobalE and PDO are the most robust baselines for demonstration ordering. Nevertheless, their worst-case accuracy falls short compared to ours, indicating that our proposed AME-ICL is more effective in selecting and ordering valuable demonstration examples to enhance ICL performance. Additionally, methods relying solely on a single characteristic, such as similarity, gradient norm, and classification entropy, may not effectively enhance prediction stability. Therefore, more precise methods for measuring sample contributions are expected to be developed. AME-ICL directly estimates the impact of each sample on ICL performance and considers the correlations among different samples, making it more reasonable and comprehensive.

AME-ICL proves highly effective in scenarios where labeled training data is unavailable, even surpassing the performance of some methods that rely on gold labels. From the results in Table II, it is apparent that when prompts are randomly sampled from the unlabeled training set (UN-Vanilla ICL), the performance is lower compared to sampling from the original labeled training set (Vanilla ICL), which is particularly pronounced in the SST-2 and AGNews datasets. These findings suggest that the input-label mapping in the prompt is crucial in ICL inference, contradicting the findings of Min et al. [31]. Encouragingly, when applying our selection method to the unlabeled training set (UN-AME-ICL), we observe not only outperformance compared to UN-Vanilla ICL but also surpassing Vanilla ICL and some other methods utilizing gold labels, such as ONESHOT and TOPPROMPTS. This suggests that input-label mapping may not always be the primary factor when valuable examples are used as demonstrations. Overall, UN-AME-ICL outperforms the baselines UN-Vanilla ICL and Vanilla ICL by 16.2% and 11.7%, respectively, on average. Moreover, compared to the best performance, AME-ICL outperforms by 5.0%. Other baselines,

TABLE II
PERFORMANCE COMPARISON BETWEEN AME-ICL AND OTHER DEMONSTRATION SELECTION APPROACHES

	SST-2		BoolQ		Subj		Scicite		AGNews		Avg. tasks
	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑	
GPTJ-6B											
Vanilla ICL†	$77.8_{11.2}$	50.8	$61.0_{3.8}$	49.7	$59.8_{8.3}$	50.1	$43.8_{7.2}$	33.6	$83.5_{3.8}$	70.4	65.2
+ CALIB†	$75.5_{9.5}$	53.6	$61.2_{3.9}$	50.4	$70.4_{7.7}$	55.7	$35.4_{2.6}$	32.8	$85.2_{2.7}$	78.0	65.5
RANDOM†	$74.6_{11.4}$	50.3	$60.0_{4.3}$	49.5	$59.9_{10.4}$	50.1	$46.4_{6.9}$	35.5	$82.5_{4.7}$	67.1	64.7
ONESHOT†	$79.6_{10.5}$	52.1	$63.8_{2.7}$	56.4	$63.3_{10.1}$	50.1	$44.8_{5.9}$	33.8	$83.3_{3.4}$	71.9	67.0
TOPPROMPTS-5†	$82.8_{8.6}$	56.0	$62.3_{3.0}$	54.3	$65.5_{9.7}$	50.1	$50.4_{6.0}$	36.9	$84.4_{3.3}$	74.3	69.1
TOPPROMPTS-10†	$78.5_{9.3}$	52.4	$61.2_{4.0}$	51.1	$65.1_{10.7}$	50.1	$49.4_{5.5}$	36.2	$85.4_{2.4}$	76.3	67.9
CONDACC†	$86.7_{5.9}$	68.2	$65.1_{1.6}$	61.1	$70.5_{10.4}$	50.2	$52.3_{4.4}$	42.0	$87.3_{2.6}$	70.5	72.4
DATAMODELS†	$86.0_{7.5}$	60.8	$65.2_{0.9}$	63.4	$69.4_{10.7}$	50.4	$56.5_{3.8}$	43.9	$86.9_{1.4}$	82.8	72.4
K-Center Greedy	$79.2_{9.9}$	50.9	$62.2_{3.5}$	53.5	$63.5_{9.9}$	50.1	$45.2_{6.7}$	35.2	$82.4_{4.4}$	69.6	66.5
GraNd	$78.9_{7.9}$	52.9	$63.1_{4.4}$	54.1	$64.0_{10.1}$	50.1	$47.0_{6.9}$	36.1	$83.3_{3.4}$	70.8	67.3
Max-Entropy	$77.5_{8.3}$	53.4	$62.8_{3.2}$	52.3	$65.1_{10.5}$	50.3	$46.3_{5.5}$	35.9	$82.7_{4.6}$	71.6	66.9
GlobalE	$88.5_{7.2}$	69.6	$64.3_{3.5}$	58.7	$75.7_{9.4}$	55.8	$53.7_{6.0}$	41.8	$80.5_{4.3}$	70.4	72.5
PDO	$88.1_{6.8}$	69.0	$65.0_{4.0}$	60.1	$74.6_{10.0}$	56.4	$54.4_{4.9}$	42.2	$80.9_{3.9}$	71.3	72.6
AME-ICL	**91.4**$_{3.5}$	**81.5**	**68.7**$_{2.0}$	**66.2**	**79.5**$_{3.2}$	**70.1**	**58.1**$_{3.1}$	**51.2**	**89.2**$_{1.0}$	**85.0**	**77.4**
UN-Vanilla ICL†	$71.0_{11.9}$	50.0	$60.8_{3.5}$	49.6	$60.1_{8.8}$	50.1	$42.0_{7.0}$	33.5	$75.1_{9.9}$	46.5	61.8
UN-ONESHOT†	$81.9_{6.3}$	68.5	$62.6_{3.3}$	55.6	$61.0_{8.7}$	50.1	$43.5_{6.7}$	33.4	$78.1_{4.2}$	69.8	65.4
UN-TOPPROMPTS-5†	$80.1_{10.5}$	56.8	$61.2_{3.3}$	51.9	$60.7_{10.0}$	50.1	$48.7_{6.9}$	33.0	$76.4_{9.8}$	53.0	65.4
UN-CONDACC†	$85.3_{6.8}$	60.5	$63.7_{2.2}$	56.0	$66.0_{10.6}$	50.1	$54.2_{3.4}$	45.9	$87.1_{1.1}$	84.6	71.3
UN-K-Center Greedy	$82.0_{7.3}$	59.4	$60.2_{3.1}$	54.9	$62.3_{9.5}$	50.1	$43.7_{5.8}$	33.6	$77.4_{6.8}$	69.5	65.1
UN-GraNd	$81.4_{6.9}$	61.0	$62.0_{2.7}$	55.0	$61.1_{10.1}$	50.2	$44.0_{6.4}$	35.2	$76.9_{5.9}$	67.0	65.1
UN-Max-Entropy	$80.7_{9.9}$	62.1	$62.2_{3.2}$	50.5	$60.9_{9.8}$	50.1	$45.3_{6.7}$	34.1	$75.5_{4.7}$	69.4	64.9
UN-PDO	$85.8_{8.5}$	65.4	$63.1_{2.9}$	55.7	$72.7_{9.6}$	55.3	$53.0_{5.9}$	41.6	$81.0_{5.3}$	70.8	71.1
UN-AME-ICL	**88.9**$_{3.9}$	**79.1**	**65.6**$_{2.2}$	**60.1**	**76.4**$_{3.6}$	**65.5**	**57.2**$_{2.7}$	**52.3**	**88.7**$_{0.9}$	**86.1**	**75.4**
OPT-13B											
Vanilla ICL†	$68.5_{14.0}$	50.0	$65.2_{5.6}$	49.7	$60.9_{10.2}$	49.8	$42.8_{3.6}$	35.0	$81.6_{5.9}$	64.2	63.8
+ CALIB†	$84.7_{6.8}$	51.7	$65.5_{4.9}$	51.8	$63.7_{8.9}$	47.9	$35.5_{1.8}$	31.2	$81.8_{4.1}$	70.7	66.2
RANDOM†	$67.7_{14.1}$	50.0	$64.7_{6.4}$	49.3	$61.2_{9.5}$	49.9	$41.2_{4.6}$	33.3	$78.0_{7.5}$	61.4	62.6
ONESHOT†	$75.6_{13.1}$	50.7	$68.3_{2.3}$	62.7	$60.5_{9.9}$	49.9	$41.9_{3.8}$	33.4	$84.2_{2.9}$	73.1	66.1
TOPPROMPTS-5†	$69.6_{14.7}$	50.0	$63.5_{6.3}$	51.0	$67.4_{12.7}$	50.0	$45.9_{4.3}$	36.0	$83.9_{3.1}$	74.0	66.1
TOPPROMPTS-10†	$72.9_{15.6}$	50.0	$65.5_{5.2}$	50.4	$68.5_{13.4}$	49.9	$44.6_{3.9}$	36.7	$84.4_{3.5}$	70.9	67.2
CONDACC†	$83.6_{9.1}$	56.1	$69.4_{2.1}$	62.8	$70.6_{11.9}$	50.0	$49.4_{3.3}$	41.1	$87.0_{1.0}$	83.6	72.0
DATAMODELS†	$81.3_{10.3}$	60.3	$69.3_{3.8}$	57.3	$63.0_{9.4}$	50.1	$46.3_{3.9}$	37.4	$85.7_{1.7}$	81.8	69.1
K-Center Greedy	$75.2_{13.8}$	50.5	$66.0_{5.7}$	54.6	$63.2_{11.2}$	49.9	$42.0_{3.8}$	32.1	$82.1_{4.2}$	74.2	65.7
GraNd	$69.1_{10.9}$	50.0	$67.1_{4.9}$	58.3	$62.9_{10.0}$	50.0	$44.8_{3.7}$	33.2	$84.2_{2.9}$	72.4	65.6
Max-Entropy	$74.2_{11.2}$	51.7	$68.0_{5.2}$	51.5	$63.0_{9.8}$	50.1	$43.1_{4.0}$	34.1	$83.9_{3.5}$	73.6	66.4
GlobalE	$87.3_{9.7}$	70.8	$67.5_{4.8}$	56.3	$65.3_{10.1}$	52.4	$46.8_{2.9}$	35.7	$83.5_{4.1}$	78.5	70.1
PDO	$90.4_{10.0}$	71.6	$68.4_{5.2}$	57.2	$69.9_{5.6}$	55.6	$47.5_{3.3}$	38.2	$84.1_{1.6}$	79.7	72.1
AME-ICL	**93.2**$_{3.3}$	**83.3**	**72.4**$_{2.0}$	**65.8**	**79.7**$_{2.9}$	**71.8**	**59.3**$_{1.5}$	**52.7**	**89.8**$_{0.8}$	**85.7**	**78.9**
UN-Vanilla ICL†	$61.6_{13.6}$	50.0	$64.8_{5.3}$	49.3	$55.8_{8.9}$	35.6	$41.9_{3.6}$	35.7	$67.3_{17.2}$	26.4	58.3
UN-ONESHOT†	$74.8_{15.6}$	50.0	$68.0_{2.5}$	59.8	$54.8_{6.2}$	47.1	$41.5_{4.1}$	33.7	$82.3_{4.5}$	64.9	64.3
UN-TOPPROMPTS-5†	$70.5_{17.0}$	50.0	$66.2_{3.4}$	54.6	$63.4_{12.3}$	48.3	$45.7_{4.7}$	33.6	$81.8_{6.9}$	51.8	65.5
UN-CONDACC†	$80.3_{12.8}$	50.0	$69.0_{2.6}$	61.5	$63.7_{11.7}$	49.9	$48.1_{4.0}$	39.2	$84.6_{3.1}$	72.5	69.2
UN-K-Center Greedy	$72.0_{14.6}$	50.0	$67.1_{3.0}$	57.9	$57.9_{9.8}$	46.0	$43.5_{4.2}$	33.6	$81.5_{4.9}$	65.4	64.4
UN-GraNd	$73.5_{12.9}$	50.1	$68.2_{2.6}$	58.1	$60.6_{10.4}$	47.3	$42.2_{4.5}$	35.1	$82.0_{6.1}$	64.7	65.3
UN-Max-Entropy	$71.0_{11.1}$	50.0	$67.8_{4.1}$	56.7	$56.4_{7.5}$	47.1	$41.5_{3.9}$	34.6	$81.8_{3.5}$	61.9	63.7
UN-PDO	$89.5_{6.9}$	64.8	$67.3_{2.8}$	56.5	$68.6_{6.8}$	53.0	$45.7_{4.0}$	38.3	$84.2_{3.0}$	69.8	71.1
UN-AME-ICL	**91.7**$_{3.4}$	**78.5**	**71.3**$_{2.6}$	**63.4**	**77.7**$_{3.8}$	**68.2**	**56.8**$_{4.2}$	**49.4**	**87.5**$_{3.2}$	**80.4**	**77.0**

The last column represents the average accuracy across all tasks. Overall, AME-ICL performs the best in terms of both average and worst accuracy. Notably, under the unlabeled setup, AME-ICL even outperforms some methods that utilize gold labels. † denotes results from [10].

such as UN-TOPPROMPTS, UN-CONDACC, and UN-PDO, perform better than UN-Vanilla ICL but notably worse than our approach.

B. Imbalanced Labels in Prompts

Previous studies [34], [57] have revealed that imbalanced class distributions in demonstrations significantly impair the performance of ICL. This section explores the impact of imbalanced labels in prompts on model performance. Alongside Vanilla ICL, we compare two methods renowned for addressing imbalanced labels: MetaICL [37] and Channel ICL [57]. We utilize the GPT-2-1.5B model and assess its performance on the SST-2 and Subj datasets. The number of demonstration examples is fixed at ten. We vary the ratio of samples in a class (e.g., "negative" in SST-2 and "objective" in Subj) within the prompts from 0.1 to 0.5. Considering both datasets entail binary classification tasks, a ratio of 0.5 indicates a balanced class distribution. Fig. 3(a) and (b) present the comparative results among Vanilla ICL, MetaICL, Channel ICL, and AME-ICL across various levels of imbalance on the SST-2 and Subj datasets. From the results, it is evident that class imbalance in the prompts can negatively affect ICL performance. Specifically, the performance of Vanilla ICL is highly susceptible to class imbalance, while MetaICL and Channel ICL improve the robustness of ICL when confronted with imbalanced class distributions. Nevertheless, AME-ICL achieves the highest accuracy among all compared methods and demonstrates high stability across various degrees of class imbalance. These results underscore the significant impact of selecting valuable samples for prompting in enhancing the stability and robustness of ICL predictions under imbalanced class distributions in prompts.

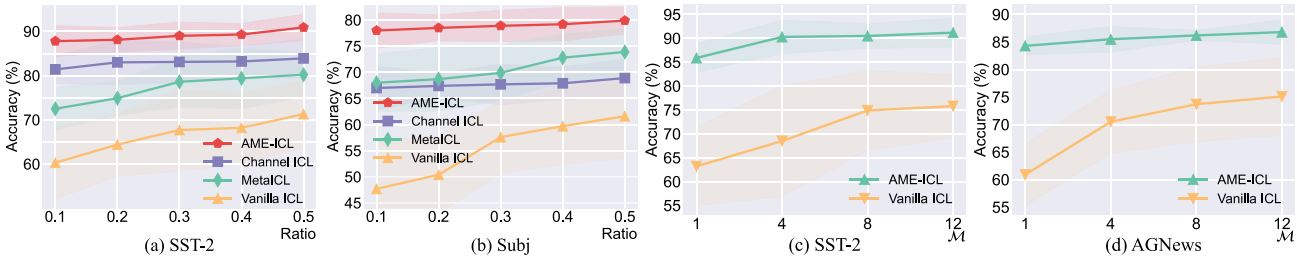

Fig. 3. (a) and (b): Accuracy comparison among Vanilla ICL, MetaICL, Channel ICL, and AME-ICL on the SST-2 and Subj datasets, where the ratios of one class (e.g., "negative" in SST-2 and "objective" in Subj) in prompts vary from 0.1 to 0.5. The GPT-2-1.5B model is utilized. (c) and (d): Accuracy comparison between Vanilla ICL and AME-ICL on the SST-2 and AGNews datasets across different numbers of demonstrations ($\mathcal{M}$) on the GPT-Neo-2.7B model.

TABLE III
COMPARISON OF ROBUST ACCURACY ON THE SST-2 AND SCICITE DATASETS UTILIZING THE GPTJ-6B AND OPT-13B MODELS

	SST-2		Scicite	
	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑
GPTJ-6B				
Vanilla ICL	$69.9_{11.6}$	45.4	$35.2_{8.0}$	30.2
TOPPROMPTS-5	$76.7_{9.2}$	52.3	$45.2_{7.6}$	32.4
CONDACC	$79.8_{7.8}$	$\underline{61.5}$	$46.6_{5.7}$	36.8
DATAMODELS	$\underline{80.0}_{6.5}$	58.4	$\underline{51.0}_{4.9}$	$\underline{37.6}$
AME-ICL	$\mathbf{87.5}_{4.1}$	**79.6**	$\mathbf{57.6}_{2.0}$	**50.4**
OPT-13B				
Vanilla ICL	$62.4_{11.2}$	46.2	$34.9_{9.7}$	30.6
TOPPROMPTS-5	$63.9_{12.0}$	47.6	$37.4_{8.5}$	32.1
CONDACC	$\underline{75.9}_{9.6}$	52.4	$\underline{44.7}_{8.1}$	$\underline{36.2}$
DATAMODELS	$75.2_{9.5}$	$\underline{54.4}$	$39.3_{7.3}$	33.7
AME-ICL	$\mathbf{90.1}_{3.2}$	**81.7**	$\mathbf{57.2}_{3.3}$	**51.2**

TABLE IV
ACCURACY COMPARISON ON IMDB AND BOOLQ CONTRAST SET, WHERE THE PROMPTS ARE COMPOSED OF THE SELECTED SST-2 AND BOOLQ TRAINING EXAMPLES, RESPECTIVELY

	IMDB		BoolQ Cst.	
	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑
GPTJ-6B				
Vanilla ICL†	$86.5_{5.7}$	63.6	$56.6_{3.0}$	50.1
TOPPROMPTS†	$87.2_{5.2}$	63.0	$56.7_{2.6}$	49.9
CONDACC†	$90.5_{1.8}$	$\underline{84.8}$	$\underline{58.9}_{1.7}$	$\underline{54.6}$
DATAMODELS†	$\underline{91.6}_{1.5}$	84.0	$57.6_{1.9}$	54.0
AME-ICL	$\mathbf{93.5}_{1.5}$	**87.6**	$\mathbf{62.8}_{1.6}$	**57.5**
OPT-13B				
Vanilla ICL†	$79.2_{12.1}$	50.1	$59.8_{2.9}$	51.6
TOPPROMPTS†	$80.5_{14.0}$	50.8	$60.3_{3.5}$	51.0
CONDACC†	$83.5_{10.8}$	54.6	$60.1_{2.1}$	$\underline{56.7}$
DATAMODELS†	$\underline{84.1}_{9.3}$	$\underline{58.9}$	$\underline{60.6}_{3.3}$	54.3
AME-ICL	$\mathbf{88.7}_{3.4}$	**76.5**	$\mathbf{66.5}_{2.0}$	**61.1**

C. Adversarial Perturbations

We validate the effectiveness of AME-ICL in enhancing the model's resilience to adversarial perturbations. Specifically, we perturb the deep features of the contextual input using (8), in which the perturbation bound ϵ is set to 0.1. Subsequently, we employ PLMs to classify the perturbed deep features and calculate the classification accuracy on the test set. The robust accuracy for the SST-2 and Scicite datasets using the GPTJ-6B and OPT-13B models is calculated. As demonstrated in Table III, AME-ICL persistently achieves the highest robust accuracy across various tasks and PLMs, showcasing its ability to enhance prediction robustness. However, the performance of all compared methods demonstrates a remarkable decline when confronted with adversarial perturbations. While CONDACC and DATAMODELS serve as strong baselines in Table II, they can not surpass our approach in terms of robust accuracy, as they primarily focus on model generalization, neglecting the models' resilience to adversarial perturbations.

D. Out-of-Distribution Tasks

We evaluate the efficacy of AME-ICL on OOD tasks, where a shift in distribution exists between the prompts and the test data. The experimental configurations follow those outlined by Chang and Jia [10]. Specifically, we employ our selection methods on a source task by sampling $\mathcal{M}$ prompts from the training data, mirroring our main experiments. Subsequently, we assess the performance on the test data of a distinct target task, ensuring a clear demarcation between the source and target tasks. For our experimental setup, we designate SST-2 and BoolQ as the source tasks, and IMDB [83] and BoolQ Contrast Set [84] as our target tasks, respectively. The findings presented in Table IV illustrate that AME-ICL achieves SOTA performance across all compared baselines on OOD tasks, indicating that rather than solely overfitting the source tasks, the selected valuable examples effectively capture patterns that can generalize well to OOD test data.

E. Cross-Model Generalization

The data values are anticipated to be transferrable across various PLMs. In such cases, employing a smaller model solely for estimating sample values becomes feasible, which can then be applied in other larger PLMs. This section explores the efficacy of valuable samples selected by a small model (i.e., GPT-2-0.8B) when utilized in the ICL phase of eight other large PLMs ranging from 1.5B to 33B (i.e., GPT-2-1.5B, GPT-Neo-2.7B, GPTJ-6B, OPT-6.7B, OPT-13B, LLaMA-33B, LLaMA-2-13B, and LLaMA-3-8B). Two datasets, SST-2 and Subj, are employed for this purpose. The experimental findings, as reported in Table V, reveal that the performance of various PLMs utilizing valuable samples selected by the GPT-2-0.8B model consistently

TABLE V
ACCURACY COMPARISON BETWEEN VANILLA ICL AND AME-ICL ON THE SST-2 AND SUBJ DATASETS, EMPLOYING NINE LARGE PLMS

	SST-2		Subj	
	Avg. std. ↑	Worst ↑	Avg. std. ↑	Worst ↑
GPT-2-0.8B				
Vanilla ICL	$57.6_{12.1}$	50.4	$57.9_{8.4}$	50.1
AME-ICL	$\mathbf{87.5}_{3.3}$	**80.8**	$\mathbf{78.1}_{2.6}$	**70.9**
GPT-2-1.5B				
Vanilla ICL	$66.3_{9.6}$	52.6	$54.2_{7.5}$	50.1
AME-ICL	$\mathbf{88.7}_{2.9}$	**80.0**	$\mathbf{79.6}_{3.4}$	**72.2**
GPT-Neo-2.7B				
Vanilla ICL	$68.9_{8.3}$	54.1	$58.2_{9.3}$	50.3
AME-ICL	$\mathbf{90.0}_{4.3}$	**81.7**	$\mathbf{82.4}_{3.0}$	**74.0**
GPTJ-6B				
Vanilla ICL	$77.8_{11.2}$	50.8	$59.8_{8.3}$	50.1
AME-ICL	$\mathbf{88.3}_{3.1}$	**80.2**	$\mathbf{77.9}_{3.0}$	**70.3**
OPT-6.7B				
Vanilla ICL	$76.5_{11.0}$	52.4	$58.3_{8.4}$	52.4
AME-ICL	$\mathbf{89.7}_{3.5}$	**80.2**	$\mathbf{77.6}_{4.1}$	**72.7**
OPT-13B				
Vanilla ICL	$68.5_{14.0}$	50.0	$60.9_{10.2}$	49.8
AME-ICL	$\mathbf{90.6}_{3.5}$	**82.1**	$\mathbf{78.3}_{2.8}$	**72.5**
LLaMA-33B				
Vanilla ICL	$93.6_{7.2}$	71.4	$83.1_{8.0}$	66.8
AME-ICL	$\mathbf{96.8}_{2.7}$	**88.6**	$\mathbf{89.9}_{3.0}$	**82.5**
LLaMA-2-13B				
Vanilla ICL	$90.5_{6.9}$	68.6	$79.8_{8.4}$	64.0
AME-ICL	$\mathbf{94.0}_{2.2}$	**86.5**	$\mathbf{87.3}_{2.1}$	**80.4**
LLaMA-3-8B				
Vanilla ICL	$93.8_{6.8}$	71.7	$84.5_{7.3}$	68.1
AME-ICL	$\mathbf{97.3}_{1.9}$	**89.0**	$\mathbf{90.8}_{2.9}$	**83.7**

All models utilize valuable demonstration examples selected by the GPT-2-0.8B model.

TABLE VI
THE FOUR GENERALIZED VALUABLE EXAMPLES CALCULATED BY AME-ICL SHARED AMONG THE NINE PLMS IN THE SST-2 DATASET

Index	Examples
1	**Review:** note that this film, like the similarly ill-timed antitrust, is easily as bad at a fraction of the budget. **Sentiment:** negative
2	**Review:** those underrated professionals who deserve but rarely receive it. **Sentiment:** positive
3	**Review:** immersed in love, lust, and sin. **Sentiment:** positive
4	**Review:** a cinematic corpse. **Sentiment:** negative

All seven models exhibit high average accuracy and low variance when utilizing these examples as demonstrations.

surpasses that of Vanilla ICL, demonstrating the effectiveness of our approach across various PLMs. Moreover, our findings suggest that the valuable training samples demonstrate transferability across various PLMs. Furthermore, we present four generalized samples in Table VI and encourage future research to investigate the distinguishing characteristics of these valuable examples. Additionally, the results manifest that our proposed AME-ICL method performs well even on gigantic PLMs, such as LLaMA-33B.

TABLE VII
ACCURACY COMPARISON ON TWO COMMONSENSE REASONING TASKS USING THE LLAMA-2-7B AND LLAMA-2-13B MODELS

	CSQA		OBQA	
	7B	13B	7B	13B
Vanilla ICL	$40.5_{5.9}$	$56.1_{6.1}$	$45.6_{4.9}$	$58.3_{5.8}$
CONDACC	$43.4_{4.7}$	$58.0_{5.5}$	$\underline{48.9}_{3.7}$	$60.7_{4.4}$
DATAMODELS	$\underline{44.0}_{4.2}$	$\underline{59.6}_{5.0}$	$48.4_{5.3}$	$\underline{60.8}_{4.6}$
AME-ICL	$\mathbf{48.7}_{2.9}$	$\mathbf{63.3}_{3.4}$	$\mathbf{52.1}_{3.5}$	$\mathbf{63.8}_{1.8}$

F. Generalization to More Complex Reasoning Tasks

To validate the applicability of our approach on more complex reasoning tasks, we conduct experiments on two commonsense reasoning datasets: CSQA [78] and OBQA [79], which are two multiple-choice commonsense question-answering tasks. Two widely used large PLMs, LLaMA-2-7B and LLaMA-2-13B, are employed in these experiments. Given that methods such as CONDACC and DATAMODELS have previously demonstrated strong performance, we consider these approaches as baseline comparisons. Other experimental settings adhere to those outlined in [29]. As shown in the results presented in Table VII, our approach consistently outperforms the baseline demonstration selection methods, highlighting its broad applicability across more complex reasoning tasks. Specifically, compared to the Vanilla ICL approach, our method achieves a 6.9% improvement. Furthermore, when compared to the best-performing baseline, our approach surpasses it by 3.7%.

G. Varying Numbers of Demonstrations

This section delves into the performance comparison between Vanilla ICL and AME-ICL utilizing different numbers of training samples as prompts. The results for the GPT-Neo-2.7B model on the SST-2 and AGNews datasets are illustrated in Fig. 3(c) and (d). As the number of demonstration examples ($\mathcal{M}$) increases, both Vanilla ICL and AME-ICL demonstrate improved performance, highlighting the importance of extensive input knowledge for the ICL inference of PLMs. Particularly noteworthy is that AME-ICL markedly enhances performance stability across varying numbers of demonstrations and consistently outperforms Vanilla ICL. This performance improvement attributed to AME-ICL is especially pronounced when $\mathcal{M}$ is smaller, indicating that the demonstrations selected by our proposed AME-ICL method encapsulate richer and more valuable knowledge of the task.

H. Varying Templates

Previous studies have highlighted that the ICL performance is sensitive to the applied templates for demonstration examples [34], [85]. To assess the performance of AME-ICL across different templates, we apply ten templates on the SST-2 dataset, as presented in Table VIII, following those outlined by Zhao et al. [34]. The OPT-13B model is utilized for this purpose. The accuracy of Vanilla ICL and AME-ICL across these ten templates is depicted in Fig. 4(a) and (b). It is observed that certain templates yield higher average performance than others.

TABLE VIII
THE TEMPLATES UTILIZED FOR EXAMINING THE INFLUENCE OF FORMATS ON THE ICL PERFORMANCE

Index	Prompt	Label names
1	Review: This movie is amazing! Answer: Positive Review: Horrific movie, don't see it. Answer:	Positive/Negative
2	Here is what our critics think for this month's films. One of our critics wrote "This movie is amazing!". Her sentiment towards the film was positive. One of our critics wrote "Horrific movie, don't see it". Her sentiment towards the film was	positive/negative
3	Review: This movie is amazing! Answer: good Review: Horrific movie, don't see it. Answer:	good/bad
4	My review for last night's film: This movie is amazing! The critics agreed that this movie was good My review for last night's film: Horrific movie, don't see it. The critics agreed that this movie was	good/bad
5	Critical reception [edit] In a contemporary review, Roger Ebert wrote "This movie is amazing!". Entertainment Weekly agreed, and the overall critical reception of the film was good. In a contemporary review, Roger Ebert wrote "Horrific movie, don't see it". Entertainment Weekly agreed, and the overall critical reception of the film was	good/bad
6	Review: This movie is amazing! Question: Is the sentiment of the above review Positive or Negative? Answer: Positive Review: Horrific movie, don't see it. Question: Is the sentiment of the above review Positive or Negative? Answer:	Positive/Negative
7	Review: This movie is amazing! Question: Did the author think that the movie was good or bad? Answer: good Review: Horrific movie, don't see it. Question: Did the author think that the movie was good or bad? Answer:	good/bad
8	Question: Did the author of the following tweet think that the movie was good or bad? Tweet: This movie is amazing! Answer: good Question: Did the author of the following tweet think that the movie was good or bad? Tweet: Horrific movie, don't see it Answer:	good/bad
9	Review: This movie is amazing! Positive Review? Yes Review: Horrific movie, don't see it. Positive Review?	Yes/No
10	This movie is amazing! My overall feeling was that the movie was good Horrific movie, don't see it. My overall feeling was that the movie was	good/bad

An example from the training set of the SST-2 dataset is provided for illustration purposes.

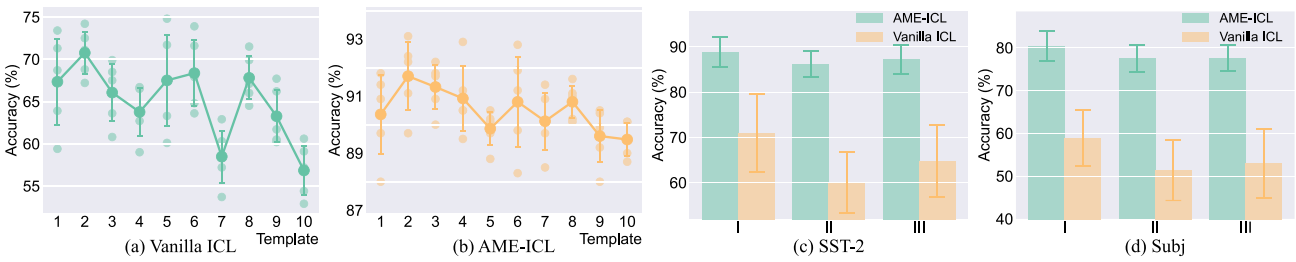

Fig. 4. (a) and (b): Accuracy comparison between Vanilla ICL and AME-ICL on the SST-2 dataset using the OPT-13B model across ten templates. (c) and (d): Accuracy comparison under three different permutation settings on the SST-2 and Subj datasets utilizing the GPT-2-1.5B model.

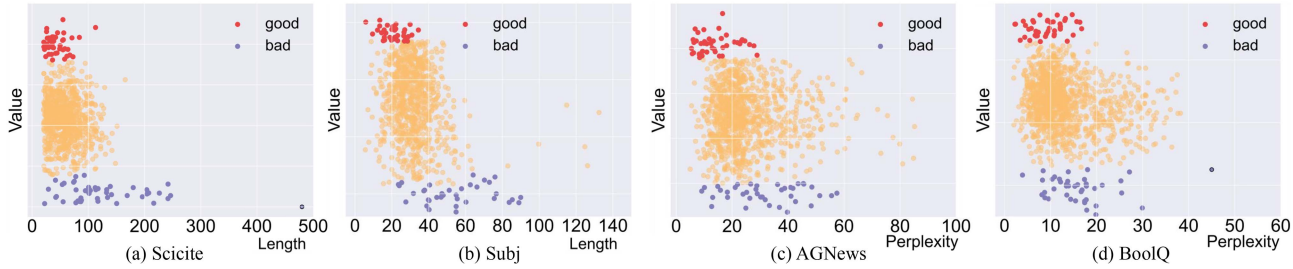

Fig. 5. (a) and (b): Data values versus sequence length on the Scicite and Subj datasets. (c) and (d): Data values versus perplexity on the AGNews and BoolQ datasets. GPT-J-6B model is utilized. Each dot corresponds to a training example and all data values are min-max normalized. High-value examples are not outliers with abnormally long lengths or high perplexities.

Nonetheless, AME-ICL consistently enhances accuracy compared to Vanilla ICL, all the while decreasing performance variance across diverse templates.

I. Varying Permutations of Demonstrations

Prior studies have highlighted that the effectiveness of ICL can be affected by the permutation of demonstration examples [6], [34]. To investigate how AME-ICL performs under various demonstration permutations, we examine the performance of AME-ICL under three permutation settings: Setting I involves arranging demonstration samples in ascending order of their total values, Setting II involves arranging them in descending order, and Setting III involves random arrangement. Subsequently, we calculate the test accuracy for each permutation on the SST-2 and Subj datasets utilizing the GPT-2-1.5B model. The results, as depicted in Fig. 4(c) and (d), suggest that AME-ICL demonstrates stability across different permutations of demonstration examples and significantly outperforms Vanilla ICL. Furthermore, optimal performance is generally achieved when the demonstrations are sorted in ascending order of total values, as samples closer to the query usually exert a greater impact on ICL prediction.

J. Analysis of Data Values

We explore the distinguishing features of selected training examples by examining them across two dimensions, namely sequence length and perplexity. Through a comparative analysis of good (i.e., high-value) and bad (i.e., low-value) training examples, we investigate how these factors influence the selection of training data. In Fig. 5, we present a scatter plot of data values against sequence length and perplexity, where each point represents a training example. The visual results in Fig. 5(a) and (b) show that low-quality samples exhibit a wide range of sequence lengths, whereas high-quality samples tend to avoid extremely long sequences. This finding is consistent with the results in CONDACC [10], suggesting that samples containing extremely long sequences during training may negatively impact ICL performance, as they tend to contain excessive redundant information.

We also calculate the perplexity of the training sample inputs. Fig. 5(c) and (d) show that samples with high values do not exhibit unusually high perplexity, suggesting that training samples with extremely high perplexity should not be selected as demonstrations, as they may contain ambiguities. However, there is no significant correlation between the data values and perplexity, indicating that perplexity alone is insufficient for identifying high-quality training samples. Nonetheless, our comparison results demonstrate that valuing training data based on the AME concept provides a more rational and accurate approach. Notably, we have confirmed that the above findings are also consistent across other tasks and PLMs beyond those visualized in Fig. 5.

TABLE IX
COMPARISON OF MSE ERRORS AMONG RIDGE REGRESSION, LASSO REGRESSION, AND ELASTIC NET

Regression	MSE	Variance
Ridge	5.71×10^{-2}	0.006
Lasso	1.43×10^{-2}	0.008
Elastic Net	$\mathbf{9.92 \times 10^{-3}}$	0.003

K. Different Linear Regressions

To determine the most suitable form of regression for our problem, we compared the mean squared error (MSE) of Ridge regression, Lasso regression, and Elastic Net. The comparison results are presented in Table IX. The results indicate that Elastic Net, which combines L_1 and L_2 regularization, achieves the lowest MSE. This is because L_1 regularization introduces sparsity, while L_2 regularization ensures parameter stability and robustness to noise. Additionally, Lasso regression outperforms Ridge regression as it is more suitable for under-determined regression problems.

Moreover, we apply these three linear regression methods to estimate data values and select the samples with the highest values as demonstrations. The comparison results on the Subj and Scicite datasets using the GPTJ-6B model are presented in Table X. As shown, Elastic Net regression, which combines both L_1 and L_2 regularization terms, outperforms the other methods. This advantage stems from Elastic Net's ability to induce sparsity while simultaneously improving parameter stability and enhancing model robustness.

Additionally, we present the cross-validation results on the SST-2 dataset to further strengthen our argument. Specifically, using the ElasticNetCV package, there are two key hyperparameters: α, which controls the overall strength of the regularization

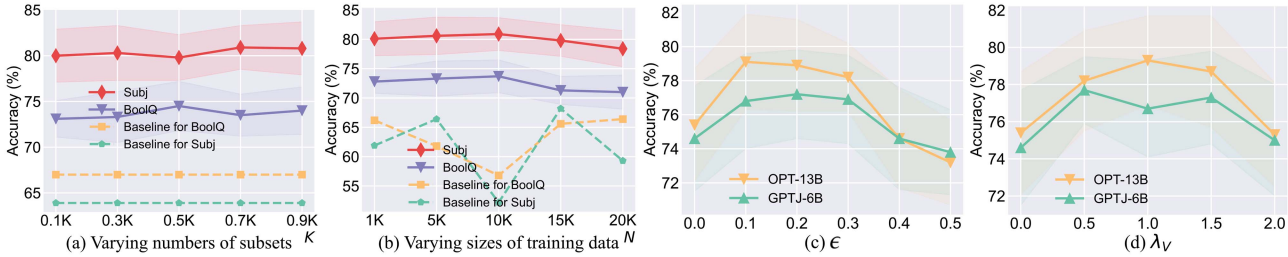

Fig. 6. (a) and (b): Accuracy of the BoolQ and Subj datasets with varying numbers of training subsets (K) and varying sizes of training data (N) on the OPT-13B model, with $\mathcal{M}$ setting to ten. (c) and (d): Sensitivity analysis for the perturbation bound ϵ and the modulating factor λ_V, using the GPTJ-6B and OPT-13B models. The average accuracy across all tasks is reported.

TABLE X
RESULTS OF ABLATION STUDIES FOR USING DIFFERENT REGRESSIONS TO CALCULATE DATA VALUES ON THE SUBJ AND BOOLQ DATASETS EMPLOYING THE GPTJ-6B MODEL

Dataset	Subj		Scicite	
Metric	Avg. std.	Worst	Avg. std.	Worst
LASSO	$\underline{77.6}_{2.8}$	$\underline{65.8}$	$\underline{57.1}_{1.9}$	47.8
Ridge	$76.3_{3.5}$	61.2	$54.0_{2.7}$	$\underline{48.5}$
Elastic Net	$\mathbf{79.5}_{3.2}$	**70.1**	$\mathbf{58.1}_{3.1}$	**51.2**

TABLE XI
CROSS-VALIDATION RESULTS ON THE SST-2 DATA USING GPTJ-6B

α	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg.
0.1	0.66×10^{-2}	0.78×10^{-2}	0.71×10^{-2}	0.74×10^{-2}	0.70×10^{-2}	3.59×10^{-2}
0.01	1.84×10^{-3}	2.09×10^{-3}	1.91×10^{-3}	2.02×10^{-3}	1.96×10^{-3}	$\mathbf{9.90 \times 10^{-3}}$
0.001	0.18×10^{-2}	0.25×10^{-2}	0.19×10^{-2}	0.23×10^{-2}	0.20×10^{-2}	1.05×10^{-2}
0.0001	1.64×10^{-2}	1.75×10^{-2}	1.69×10^{-2}	1.74×10^{-2}	1.72×10^{-2}	8.54×10^{-2}

The optimal performance is achieved with $\alpha = 0.01$.

terms, and $l1\_ratio$, which determines the relative contribution of the L_1 and L_2 regularization terms. The value of $l1\_ratio$ adopts the default settings (i.e., 0.5) in the ElasticNetCV package. Moreover, the results of the five-fold cross-validation experiments for different values of α are summarized in Table XI. As we can see, the best performance is achieved when $\alpha = 0.01$. In this context, the coefficients for the two regularization terms are 0.005 and 0.01, respectively.

L. Efficiency and Scalability

We explore the efficiency of AME-ICL to highlight its scalability. The additional time consumption of AME-ICL primarily arises from utilizing prompts generated from diverse training subsets for inference. Consequently, the time required increases proportionally with the number of prompts constructed. As depicted in Fig. 6(a), with a fixed training size of 1,000, satisfactory results can be attained with just 0.1 K subsets. Furthermore, an increase in the number of subsets only leads to minor fluctuations and improvements in performance. These findings suggest that AME-ICL can effectively select the most valuable samples with a small number of prompts, owing to its consideration of sparsity.

Additionally, as shown in Fig. 6(b), when the size of the training data becomes large (i.e., 15 K and 20 K), there is only a slight decline in ICL performance, which still significantly outperforms the baseline. Notably, other demonstration valuation methods, such as CONDACC and DATAMODELS, consume hundreds of GPU hours due to the need for inference with numerous prompts (exceeding 50 K prompts for a training set size of 1,000). Nevertheless, AME-ICL significantly enhances efficiency compared to baseline methods such as CONDACC and DATAMODELS, while still achieving substantial performance improvements. Specifically, running the OPT-13B model on BoolQ takes over 500 GPU hours on an RTXA6000 GPU for CONDACC and DATAMODELS, whereas our method consumes less than 10 hours. Fig. 6(b) also indicates that as the size of the training data increases, the number of sampled subsets is expected to increase accordingly, aiming to estimate the data values more accurately. Furthermore, having verified the transferability of data values across different model sizes, we can directly employ small PLMs for data valuation and subsequently transfer the valuable samples to large models.

TABLE XII
RESULTS OF ABLATION STUDIES FOR DIFFERENT CONFIGURATIONS OF DATA VALUES ON THE SUBJ AND BOOLQ DATASETS

Values		Subj		BoolQ	
$\boldsymbol{V}^g$	$\boldsymbol{V}^r$	Avg. std.	Worst	Avg. std.	Worst
✓	✗	$75.9_{3.2}$	67.3	$66.9_{3.1}$	63.6
✗	✓	$74.2_{4.0}$	64.9	$66.2_{3.3}$	61.1
✓	✓	$\mathbf{77.8}_{3.2}$	**70.1**	$\mathbf{68.7}_{2.0}$	**66.2**

M. Ablation and Sensitivity Studies

Ablation studies and sensitivity tests are conducted on AME-ICL to gain deeper insights into the impact of each of its components. Initially, we scrutinize the performance when considering individual generalization and robustness values. Specifically, we conduct three sets of experiments on the Subj and BoolQ datasets using the GPTJ-6B model: considering only the generalization value ($\boldsymbol{V}^g$), only the robustness value ($\boldsymbol{V}^r$), and both values combined. The results, as reported in Table XII, reveal that simultaneously considering both values yields optimal performance, underscoring the significance of both model generalization and robustness during ICL inference. Moreover, focusing solely on $\boldsymbol{V}^g$ generally yields superior results compared to solely focusing on $\boldsymbol{V}^r$.

TABLE XIII
PERFORMANCE COMPARISON UNDER VARYING SAMPLING DISTRIBUTIONS OF TRAINING SUBSETS

	BoolQ		AGNews	
	Avg. std.	Worst	Avg. std.	Worst
Setting I	$\mathbf{68.7}_{2.0}$	66.2	$\mathbf{89.2}_{1.0}$	**85.0**
Setting II	$68.3_{1.5}$	**66.5**	$88.8_{1.7}$	84.9
Setting III	$66.6_{1.9}$	64.7	$88.3_{2.2}$	83.9

Moreover, we conduct experiments to examine the effect of different sampling distributions of training subsets on ICL performance. Three settings are considered: Setting I: $\mathcal{P} = \text{Uniform}\{0.1, 0.2, 0.3, 0.4\}$, Setting II: $\mathcal{P} = \text{Uniform}\{0.2, 0.4, 0.6, 0.8\}$, and Setting III: $\mathcal{P} = \text{Uniform}\{0.5, 0.6, 0.7, 0.8\}$. Using datasets BoolQ and AGNews, we perform experiments on the GPTJ-6B model. The results, shown in Table XIII, reveal that Settings I and II yield favorable outcomes while Setting III performs poorly. This occurs because, when constructing prompts with relatively larger subsets, compared to smaller subsets, the ability to capture the performance variations of ICL as each individual data point is added becomes less effective. As a result, the influence of individual data points tends to be overshadowed.

Sensitivity tests for the hyperparameters in AME-ICL have also been conducted. Two key hyperparameters are considered: the perturbation bound ϵ used when calculating the robustness utility, and the modulating factor λ_V between the two values. The average accuracy across all tasks on the OPT-13B and GPTJ-6B models is calculated. As illustrated in Fig. 6(c) and (d), the performance of AME-ICL remains stable when ϵ falls within the set of $\{0.1, 0.2, 0.3\}$ and λ_V is selected from $\{0.5, 1.0, 1.5\}$. Therefore, in real-world applications, hyperparameter values can be selected from these stable sets.

VI. CONCLUSION AND FUTURE WORK

This study introduces a novel method, namely AME-ICL, to identify valuable training samples for prompting in ICL. Two types of data values pertaining to model generalization and robustness are calculated. Subsequently, samples with the highest combined values are selected and ordered to construct task-specific prompts. Our AME-ICL method is intuitive and straightforward to implement, enabling seamless integration with various PLMs. The extensive experiments demonstrate that AME-ICL consistently outperforms previous demonstration selection approaches in terms of both average and worst-case accuracy. Moreover, it significantly enhances the stability and robustness of ICL predictions.

Given the promising results of AME-ICL, there are several avenues that warrant further exploration. First, future research could conduct a more comprehensive analysis of the characteristics of valuable samples to establish guidelines for selecting or creating optimal samples. Moreover, our framework is highly scalable and easily adaptable to handle other tasks by substituting the metrics for utilities with alternative indicators. For instance, in generation tasks, we can utilize BLEU and ROUGE metrics. Third, considering that the number of available samples for the ICL process may vary over time, investigating the incremental or decremental valuation of training data would be both intriguing and meaningful.

REFERENCES

[1] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1413–1430, Apr. 2024.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 4171–4186.

[3] F. Petroni et al., "Language models as knowledge bases?," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2019, pp. 2463–2473.

[4] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[5] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[6] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 8086–8098.

[7] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, Pp. 24824–24837.

[8] M. Li et al., "In-context learning with many demonstration examples," 2023, *arXiv:2302.04931*.

[9] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2022, pp. 2655–2671.

[10] T.-Y. Chang and R. Jia, "Data curation alone can stabilize in-context learning," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2023, pp. 8123–8144.

[11] T. Song, "Provenance-driven data curation workflow analysis," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 45–50.

[12] Z. Shang et al., "Democratizing data science through interactive curation of ML pipelines," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2019, pp. 1171–1188.

[13] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A Big Data-AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, Apr. 2021.

[14] R. Y. Wang, V. C. Storey, and C. P. Firth, "A framework for analysis of data quality research," *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 4, pp. 623–640, Aug. 1995.

[15] R. Das et al., "Case-based reasoning for natural language queries over knowledge bases," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 9594–9611.

[16] R. Shin et al., "Constrained language models yield few-shot semantic parsers," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2021, pp. 7699–7715.

[17] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?," in *Proc. Deep Learn. Inside Out: Workshop Knowl. Extract. Integr. Deep Learn. Archit.*, 2021, pp. 100–114.

[18] Y. Zhang, S. Feng, and C. Tan, "Active example selection for in-context learning," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2022, pp. 9134–9148.

[19] H. Su et al., "Selective annotation makes language models better few-shot learners," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–13.

[20] L. Davezies, X. D'Haultfoeuille, and L. Laage, "Identification and estimation of average marginal effects in fixed effects logit models," 2021, *arXiv:2105.00879*.

[21] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[22] G. Lecué and S. Mendelson, "Regularization and the small-ball method I: Sparse recovery," *Ann. Statist.*, vol. 46, no. 2, pp. 611–641, 2018.

[23] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc. B*, vol. 67, no. 2, pp. 301–320, 2005.

[24] N. Wies, Y. Levine, and A. Shashua, "The learnability of in-context learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 36637–36651.

[25] R. Agarwal et al., "Many-shot in-context learning," in *Proc. ICML 2024 Workshop In-Context Learn.*, 2024, pp. 76930–16966.

[26] Q. Huang et al., "PRODIGY: Enabling in-context learning over graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 16302–16317.

[27] C. Qin, A. Zhang, C. Chen, A. Dagar, and W. Ye, "In-context learning with iterative demonstration selection," in *Proc. Findings Assoc. Comput. Linguist.*, 2023, pp. 7441–7455.
[28] Z. Jiang, Y. Zhang, C. Liu, J. Zhao, and K. Liu, "Generative calibration for in-context learning," in *Proc. Findings Assoc. Comput. Linguist.*, 2023, pp. 2312–2333.
[29] H. Zhang et al., "A study on the calibration of in-context learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2024, pp. 6118–6136.
[30] J. Yan, J. Xu, C. Song, C. Wu, Y. Li, and Y. Zhang, "Understanding in-context learning from repetitions," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–13.
[31] S. Min et al., "Rethinking the role of demonstrations: What makes in-context learning work?," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2022, pp. 11048–11064.
[32] L. Qu, S. Wu, H. Fei, L. Nie, and T.-S. Chua, "LayoutLLM-T2I: Eliciting layout guidance from LLM for text-to-image generation," in *Proc. ACM Int. Conf. Multimedia*, 2023, pp. 643–654.
[33] Y. Fei, Y. Hou, Z. Chen, and A. Bosselut, "Mitigating label biases for in-context learning," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2023, pp. 14014–14031.
[34] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12697–12706.
[35] T. Nguyen and E. Wong, "In-context example selection with influences," 2023, *arXiv:2302.11042*.
[36] J. Wei et al., "Larger language models do in-context learning differently," 2023, *arXiv:2303.03846*.
[37] S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi, "MetaiCL: Learning to learn in context," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2022, pp. 2791–2809.
[38] P. Shi, R. Zhang, H. Bai, and J. Lin, "XRICL: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-SQL semantic parsing," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2022, pp. 5248–5259.
[39] X. Li et al., "Unified demonstration retriever for in-context learning," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2023, pp. 4644–4668.
[40] I. Levy, B. Bogin, and J. Berant, "Diverse demonstrations improve in-context compositional generalization," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2023, pp. 1401–1422.
[41] S. Agrawal, C. Zhou, M. Lewis, L. Zettlemoyer, and M. Ghazvininejad, "In-context examples selection for machine translation," in *Proc. Annu. Meet. Assoc. Comput Linguist.*, 2023, pp. 8857–8873.
[42] P. Ren et al., "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–40, 2021.
[43] K. Margatina, T. Schick, N. Aletras, and J. Dwivedi-Yu, "Active learning principles for in-context learning with large language models," in *Proc. Findings Assoc. Comput. Linguist.*, 2023, pp. 5011–5034.
[44] J. Pei, "A survey on data pricing: From economics to data science," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4586–4608, Oct. 2022.
[45] Y. Kwon and J. Zou, "Beta shapley: A unified and noise-reduced data valuation framework for machine learning," in *Proc. Int. Conf. Mach. Learn. Res.*, 2022, pp. 8780–8802.
[46] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2242–2251.
[47] H. A. Just et al., "LAVA: Data valuation without pre-specified learning algorithms," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–12.
[48] J. Yoon, S. Arik, and T. Pfister, "Data valuation using reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10842–10851.
[49] Y. Kwon and J. Zou, "Data-OOB: Out-of-bag estimate as a simple and efficient data value," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 18135–18152.
[50] K. F. Jiang, W. Liang, J. Zou, and Y. Kwon, "OpenDataVal: A unified benchmark for data valuation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 28624–28647.
[51] J. T. Wang and R. Jia, "Data Banzhaf: A robust data valuation framework for machine learning," in *Proc. Int. Conf. Mach. Learn. Res.*, 2023, pp. 6388–6421.
[52] M. Stoian, "Fast joint Shapley values," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2023, pp. 285–287.
[53] X. Luo, J. Pei, C. Xu, W. Zhang, and J. Xu, "Fast Shapley value computation in data assemblage tasks as cooperative simple games," in *Proc. ACM Manage. Data*, vol. 2, no. 1, pp. 1–28, 2024.
[54] R. Jia et al., "Towards efficient data valuation based on the Shapley value," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1167–1176.
[55] J. Lin, A. Zhang, M. Lécuyer, J. Li, A. Panda, and S. Sen, "Measuring the effect of training data on deep learning predictions via randomized experiments," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 13468–13504.
[56] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry, "Datamodels: Predicting predictions from training data," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 9525–9587.
[57] S. Min, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Noisy channel language model prompting for few-shot text classification," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2022, pp. 5316–5330.
[58] Z. Han, Y. Hao, L. Dong, Y. Sun, and F. Wei, "Prototypical calibration for few-shot learning of language models," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–11.
[59] X. Li and X. Qiu, "Finding support examples for in-context learning," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2023, pp. 6219–6235.
[60] E. Perez, D. Kiela, and K. Cho, "True few-shot learning with language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 11054–11070.
[61] N. Egami and K. Imai, "Causal interaction in factorial experiments: Application to conjoint analysis," *J. Amer. Statist. Assoc.*, vol. 114, no. 526, pp. 529–540, 2019.
[62] J. D. Angrist and J.-S. Pischke, *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[63] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.
[64] B. Williamson and J. Feng, "Efficient nonparametric statistical inference on population feature importance using Shapley values," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10282–10291.
[65] J. Kossen, Y. Gal, and T. Rainforth, "In-context learning learns label relationships but is not conventional learning," in *Proc. Int. Conf. Learn. Representations*, 2024, pp. 1–14.
[66] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 billion parameter autoregressive language model," 2022, Art. no. 8. [Online]. Available: https://github.com/kingoflolz/mesh-transformer-jax
[67] S. Zhang et al., "OPT: Open pre-trained transformer language models," 2022, *arXiv:2205.01068*.
[68] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," in *Proc. OpenAI Blog*, 2019, pp. 1–12.
[69] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, "GPT-neo: Large scale autoregressive language modeling with mesh-tensorflow," 2022, Art. no. 5297715. [Online]. Available: https://doi.org/10.5281/zenodo
[70] H. Touvron et al., "Llama: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.
[71] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
[72] A. Dubey et al., "The Llama 3 herd of models," 2024, *arXiv:2407.21783*.
[73] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2013, pp. 1631–1642.
[74] C. Clark et al., "BoolQ: Exploring the surprising difficulty of natural yes/no questions," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 2924–2936.
[75] A. Cohan, W. Ammar, M. Van Zuylen, and F. Cady, "Structural scaffolds for citation intent classification in scientific publications," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 3586–3596.
[76] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2004, pp. 271–278.
[77] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[78] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4149–4158.
[79] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? A new dataset for open book question answering," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 2381–2391.
[80] Z. Xu, D. Cohen, B. Wang, and V. Srikumar, "In-context example ordering guided by label distributions," in *Proc. Findings Assoc. Comput. Linguist.*, 2024, pp. 2623–2640.

[81] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.
[82] M. Paul, S. Ganguli, and G. K. Dziugaite, "Deep learning on a data diet: Finding important examples early in training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 20596–20607.
[83] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 142–150.
[84] M. Gardner et al., "Evaluating models' local decision boundaries via contrast sets," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 1307–1323.
[85] T. Sorensen et al., "An information-theoretic approach to prompt engineering without ground truth labels," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2022, pp. 819–862.

Xiaoling Zhou received the BSc degree in mathematics from Tiangong University, Tianjin, China, in 2020, and the MSc degree in mathematics from the Center for Applied Mathematics, Tianjin University, Tianjin, China, in 2023. She is currently working toward the PhD degree with the National Engineering Research Center for Software Engineering, Peking University, Beijing, China. Her research interests include deep learning, large language models, and adversarial training.

Wei Ye received the BS degree in computer science and technology from the University of Electronic Science and Technology of China, Sichuan, China, in 2006, and the PhD degree from the School of Electronics Engineering and Computer Science of Peking University, Beijing, China, in 2011. He is currently an associate professor and doctoral supervisor with the National Engineering Research Center for Software Engineering, Peking University, Beijing, China. His research interests include artificial intelligence and deep learning.

Zhemg Lee received the BS degree from the School of Physical Science and Technology, Tiangong University, Tianjin, China, in 2020, and the MS degree in electronic and information engineering from Tianjin University, Tianjin, China, in 2023. He is currently working toward the PhD degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include deep learning, AI for science, and optical computing.

Lei Zou is a professor with the Wangxuan Institute of Computer Technology, Peking University. He is also a faculty member with the National Engineering Laboratory for Big Data Analysis and Applications (Peking University) and the Center for Data Science of Peking University. His research interests include graph databases and semantic data management.

Shikun Zhang received the bachelor's, master's, and PhD degrees in computer software from Peking University, Beijing, China, in 1990, 1993, and 2000, respectively. He is currently a full professor with the National Engineering Research Center for Software Engineering, Peking University, Beijing, China. His research interests include artificial intelligence, software security, and software engineering.