Valuing Training Data via Causal Inference for In-Context Learning

Submission Id: 167

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.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Supervised learning by regression**; • **Networks** → **Network performance analysis**.

KEYWORDS

In-context learning, Data valuation, Causal inference, Average marginal effect, Elastic Net regression

1 INTRODUCTION

The remarkable linguistic capabilities and extensive world knowledge embedded in large pre-trained language models (PLMs) [2, 8, 40, 42] 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 [30, 59], as well as reducing computational costs [26, 45].

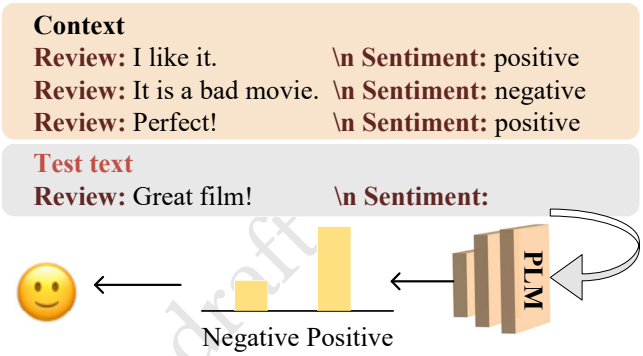

Figure 1: Illustration of ICL for sentiment classification.

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 may result in poor performance [3, 30]. Therefore, data curation [48, 52] 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) [3, 6, 50], training dense retrievers [29, 45], and active learning-based approaches [55, 66]. However, these methods can not effectively 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. Furthermore, most of these approaches require substantial resources for either training or inference [45, 66], thereby impeding the efficiency of ICL.

In this study, we tackle this challenge for the first time from the perspective of training data valuation through causal inference. 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 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) [7] to evaluate the contribution of each training sample and define it as the expected marginal effect on the inference performance during the ICL process. 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.

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, 68], to tackle this regression problem. This approach notably improves computational efficiency by sampling fewer training subsets (fewer than the total number of training data). 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 approach, named AME-ICL[1], 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 across five classification tasks and seven PLMs, showcasing that AME-ICL consistently attains 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.

[1]The code for AME-ICL has been released on an anonymized GitHub repository: https://anonymous.4open.science/r/AME-ICL-3198.

2 RELATED WORK

2.1 In-Context Learning

Brown et al. [2] 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 [14, 34, 35] have extended and refined this approach, resulting in promising outcomes across a spectrum of tasks. For instance, Min et al. [35] designed a Meta-ICL framework, where a PLM is fine-tuned to perform ICL on a vast array of training tasks distinct from the target one. This approach enables the model to more efficiently learn a new task within the context at test time. Moreover, Zhao et al. [67] suggested calibrating the predictions of PLMs, effectively reducing the performance variance across different prompt choices. Additionally, Min et al. [34] proposed a noisy channel approach for ICL prompting in few-shot text classification. This method calculates the conditional probability of the input given the label and is thus required to elucidate every word in the input. Recently, significant progress has also been made in the understanding of ICL. For example, Saunshi et al. [46] proposed that conditioning on a prompt can render the task of predicting the next word linearly separable. Levine et al. [24] introduced a pre-training scheme theoretically motivated by the bias of ICL, resulting in notable improvements. Additionally, Xie et al. [62] revealed that ICL occurs when the model identifies a shared latent concept among examples in a prompt. Furthermore, Min et al. [36] demonstrated that the model does not heavily rely on the ground truth input-label mapping provided in the demonstrations.

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) or labels appearing later in a prompt (recency bias) [67]. The latter implies that optimizing the order of demonstration examples could lead to performance improvements [30]. Moreover, the structure of the prompt template, or the format in which the example is presented, also plays a significant role [34]. Additional research has revealed that the accuracy of input-label mapping has minimal impact [35], while the diversity of examples is of greater importance [55].

2.2 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, Liu et al. [29] employed a semantic embedder to retrieve the most similar examples given a query. Rubin et al. [45] and Shi et al. [49] trained the prompt retriever based on feedback from PLMs for semantic parsing. Moreover, Wu et al. [61] utilized Sentence-BERT [43] for relevant sample retrieval and introduced an information-theoretic-driven criterion for sorting them. Furthermore, Levy et al. [25] posited that diverse demonstrations would benefit ICL inference. 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.

Another line to retrieve prompts involves active learning [44]. For example, Zhang et al. [66] 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, Su et al. [55] introduced a graph-based annotation method known as vote-k. They employed Sentence-BERT to retrieve relevant examples from the annotated set for ICL. However, these methods consume significant computational resources and are sensitive to noise. This study explores a new path for prompt retrieval. Specifically, we utilize the AME concept to assess the contribution of each training sample to the generalization and robustness performance of ICL predictions, and then select the most valuable ones to construct prompts. Our method directly quantifies the impact of each training sample on ICL inference and effectively considers the correlations among individual demonstration examples. Furthermore, unlike methods that rely on training dense extractors and active learning techniques, our approach involves solving a linear regression and constructing task-level prompts, which is more efficient and straightforward.

2.3 Data Valuation

The objective of data valuation is to assess the individual contribution of each data point to model behavior. 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 [20, 28], gradient-based methods [17, 18], importance weight-based methods [63], and out-of-bag (OOB) estimation-based methods [21]. 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 [16], Data Banzhaf [58], and a range of Shapley value-based approaches [32, 54] such as Data Shapley [12], Beta Shapley [20], and AME [7]. Notably, this type of method typically entails training different models on extensive training subsets. 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 [18], datamodels [15], and LAVA [17]. 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 [63] 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 [21], which computes data point contribution using out-of-bag accuracy.

Our method falls within the category of marginal contribution-based approaches considering its direct measurement and effectiveness. Specifically, we leverage AME to assess the contribution of each training sample. However, unlike previous AME methods [7, 23, 28], we innovatively focus on ICL, where training examples act as prompts without updating parameters for large PLMs. As a result, the need to train different models on diverse training subsets is eliminated. Additionally, we estimate the AME values using Elastic Net regression, chosen for its capability to handle sparse solutions and ensure parameter stability. Furthermore, both model generalization and robustness performance are considered as utilities to enable a more comprehensive data valuation.

3 METHODOLOGY

3.1 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 [10, 13, 27, 34], this study focuses on the classification task. Specifically, considering a PLM G, given an input text $\boldsymbol{x}$ and a candidate answer set $L = \{y_1, y_2, \cdots, y_{|L|}\}$ with $|L|$ classes, we aim to predict the answer $\hat{y}$ for $\boldsymbol{x}$ based on $\mathcal{M}$ selected valuable training examples: $C = \{e_1, e_2, \cdots, e_{\mathcal{M}}\}$, 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:

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

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

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

The prediction accuracy for the test set Acc^{te} is utilized to evaluate the performance of our approach, 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 [3]: 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 [39], 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.

3.2 Training Data Valuation

3.2.1 ***Utility Definition.*** Our objective in data valuation is to explore the impact of each training point on the performance of ICL

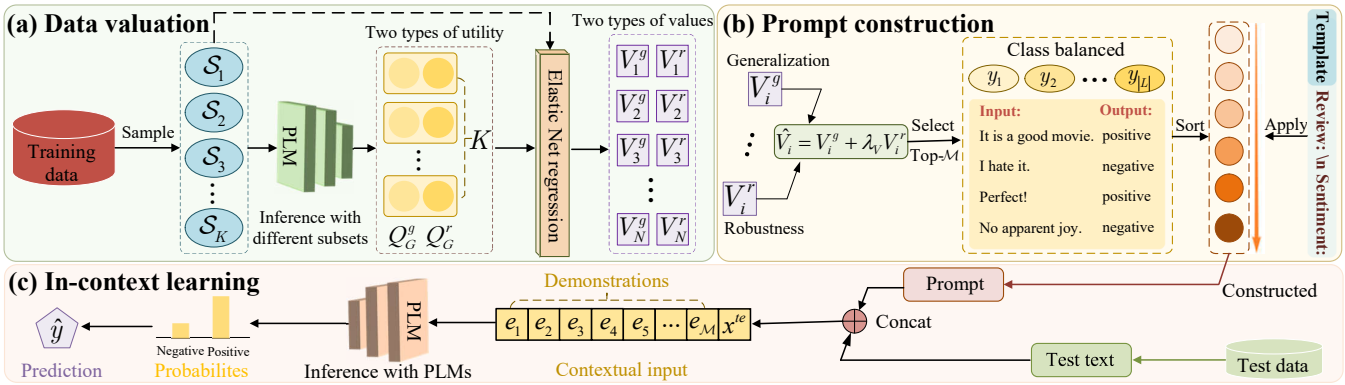

Figure 2: The pipeline of AME-ICL consists of three main steps: training data valuation, prompt construction, and ICL inference. Firstly, we compute the values of all training data in terms of both the generalization and robustness performance of ICL inference. Subsequently, we select the most valuable 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.

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 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:

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

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:

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

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

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

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.

3.2.2 ***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 domain [9], we average the marginal contributions of including data point e_i across various training subsets. Consequently, the AME value of e_i (V_i) is defined as its expected marginal effect [23] 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 $\mathcal{N}^{tr}$. In implementation, 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].

[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 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 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**
  $\mathcal{S}_k \leftarrow \{\}$;
  $p \sim \mathcal{P}$;
  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* 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);
  $\boldsymbol{Y}[k] \leftarrow Q_G(\mathcal{S}_k)$;
$\boldsymbol{V}_{EN} = \arg\min_{\boldsymbol{V}\in\mathbb{R}^N} \left((\boldsymbol{Y} - <\boldsymbol{V}, \boldsymbol{X}>)^2 + \alpha||\boldsymbol{V}||_1 + \beta||\boldsymbol{V}||_2\right)$

To simultaneously calculate the AME values for all training data, we reframe the estimation of all $\boldsymbol{V}$ values as a specific linear regression problem, following the approach of Lin et al. [28]. Inspired by randomized experiments, we initiate by generating K subsets of the training data, denoted as $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_K$. 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 AME values of all training points.

To enhance efficiency, 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 underdetermined regression problem, as the number of equations is fewer than the number of variables [31, 60]. Additionally, due to the input length limitations of PLMs, only a limited number of samples can be selected as demonstrations. Consequently, we leverage sparsity

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 $C = \{e_1, e_2, \cdots, e_{\mathcal{M}}\}$;
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 C;
  $\mathcal{A}_i \leftarrow \sum_{j=1}^{\mathcal{B}} \mathbb{I}(\hat{y}_j(C) = y_j)$;
$Acc^{te} \leftarrow \frac{1}{\mathcal{B}\times\lfloor \frac{|\mathcal{D}^{te}|}{\mathcal{B}} \rfloor} \sum_{i=1}^{\lfloor \frac{|\mathcal{D}^{te}|}{\mathcal{B}} \rfloor} \mathcal{A}_i$

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.

3.3 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 [19, 67] 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

Table 1: The templates and label mappings across different tasks. 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.

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

closest to the query are prioritized as the most valuable ones. Finally, the constructed prompt 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.

4 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. These scenarios include imbalanced labels in prompts, adversarial perturbations, and OOD tasks. Additionally, we assess the transferability of data values across different PLMs and investigate the performance stability across varying numbers, templates, and permutations of demonstration examples. 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 the impact of each of its components.

4.1 Setups

Seven PLMs, ranging in size from 0.8B to 33B, are employed to showcase the adaptability of AME-ICL across different model sizes. The primary comparison experiments involve two PLMs: GPTJ-6B [57] and OPT-13B [64]. Further analytical experiments are conducted on GPT-2-0.8B, GPT-2-1.5B [41], GPT-Neo-2.7B [1], OPT-6.7B [64], and LLaMA-33B [56]. Following previous research [3, 13, 34], our experiments are conducted on five classification tasks: SST-2 [51], BoolQ [4], Subj [37], Scicite [5], and AGNews [65]. Table 1 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. During data evaluation, the number of subsets K is set to 100; ten random permutations are considered for each subset. Additionally, each experiment is repeated using five random seeds. 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 $\hat{\boldsymbol{V}}$. But we also explore other permutations in the analytical experiments. For each task, a specific template is utilized for inference, as presented in Table 1. Additionally, we examine the impact of different templates on the performance of AME-ICL following those outlined by Zhao et al. [67], which are listed in Table 7. ElasticNetCV is utilized to determine the optimal values of α and β through cross-validation, using default hyperparameter settings. As for other hyperparameters in AME-ICL, we select the perturbation bound ϵ from the set $\{0.1, 0.2, 0.3\}$, and choose the parameter λ_V from the set $\{0.5, 1.0, 1.5\}$.

4.2 Evaluation and Baselines

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 [3], 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 $\mathcal{D}^{te}$. In this setting, both the **average accuracy** and **standard deviation** are reported.

As for the compared baselines, we first benchmark AME-ICL against seven baseline methods introduced by Chang and Jia [28]. **Vanilla ICL** randomly selects demonstration examples from the entire training set. **CALIB**, building upon Vanilla ICL, incorporates calibration techniques [67] to mitigate biases towards specific labels

Table 2: Performance comparison between AME-ICL and other demonstration selection approaches. 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 [3].

	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
AME-ICL	**89.8**$_{3.5}$	**81.5**	**68.7**$_{2.0}$	**66.2**	**77.8**$_{3.2}$	**70.1**	**58.1**$_{3.1}$	**51.2**	**89.2**$_{1.0}$	**85.0**	**76.7**
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-AME-ICL	**87.9**$_{3.8}$	**79.1**	**65.6**$_{2.2}$	**60.1**	**72.4**$_{3.6}$	**65.5**	**57.2**$_{2.7}$	**52.3**	**88.7**$_{0.9}$	**86.1**	**74.4**
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
AME-ICL	**91.2**$_{3.1}$	**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**	**78.5**
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-AME-ICL	**87.3**$_{3.2}$	**78.5**	**71.3**$_{2.6}$	**63.4**	**76.7**$_{3.8}$	**68.2**	**56.8**$_{4.2}$	**49.4**	**87.5**$_{3.2}$	**80.4**	**75.9**

in PLMs. **RANDOM** randomly selects a balanced training subset consisting of twenty examples and then chooses demonstration examples from this subset. **ONESHOT** takes a different approach by 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** [28] scores a training example based on its average ICL accuracy on the validation set when combined with random training examples. **DATAMODELS** [28] trains a datamodel to predict the PLM's output for each sample in $\mathcal{D}^{dev}$. Besides, we compare AME-ICL with three metric-based selection approaches: **K-Center Greedy** [47], which assumes that training samples close in feature space have similar properties, thereby selecting samples with high similarities; **GraNd** [38], which selects the most informative examples based on the gradient norm expectations of samples; and **Max-Entropy** [66], which greedily selects examples to maximize classification entropy.

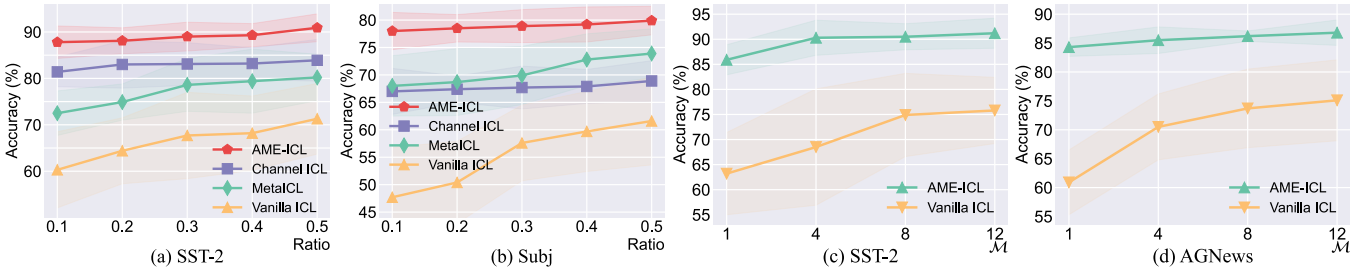

Figure 3: (a) and (b): Accuracy comparison among Vanilla ICL, MetaICL, Channel ICL, and AME-ICL on the SST-2 and Subj datasets, where the ratios of one class (e.g., "negative" in SST-2 and "objective" in Subj) in prompts vary from 0.1 to 0.5. The GPT-2-1.5B model is utilized. (c) and (d): Accuracy comparison 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 3: 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	$79.8_{7.8}$	61.5	$46.6_{5.7}$	36.8
DATAMODELS	$80.0_{6.5}$	58.4	$51.0_{4.9}$	37.6
AME-ICL	**87.5**$_{4.1}$	**79.6**	**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	$75.9_{9.6}$	52.4	$44.7_{8.1}$	36.2
DATAMODELS	$75.2_{9.5}$	54.4	$39.3_{7.3}$	33.7
AME-ICL	**90.1**$_{3.2}$	**81.7**	**57.2**$_{3.3}$	**51.2**

Finally, we extend AME-ICL and these compared baseline methods to the unlabeled setup, denoted by the UN-prefix.

5 EXPERIMENTAL FINDINGS

5.1 Main Comparison Results

AME-ICL consistently demonstrates the highest average accuracy with low standard variance, highlighting its ability to enhance the generalization capability of PLMs. Table 2 presents the test set accuracy achieved using different demonstration selection approaches. AME-ICL establishes itself as the SOTA method among all compared techniques, showcasing its exceptional generalization ability. Moreover, it exhibits minimal variance across different random seeds, underscoring its capacity to enhance performance stability. Overall, AME-ICL exhibits a 14.1% 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 5.2%. 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 the strongest baselines. Nevertheless, AME-ICL notably outperforms these two approaches, suggesting that our proposed data valuation approach is more accurate and effective, thus facilitating the selection of demonstration samples that enhance the overall inference performance.

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 2, AME-ICL's worst accuracy exceeds that of the best-performing method among the baselines by 11.6% on average. Furthermore, compared to Vanilla ICL, AME-ICL exhibits an improvement of 23.3%. 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 emerge as the most robust baselines. Nevertheless, their worst-case accuracy falls short compared to ours, indicating that our proposed AME-ICL is more effective in selecting 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.

Table 4: 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†	$90.5_{1.8}$	84.8	$58.9_{1.7}$	54.6
DATAMODELS†	$91.6_{1.5}$	84.0	$57.6_{1.9}$	54.0
AME-ICL	**93.5**$_{1.5}$	**87.6**	**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	$60.1_{2.1}$	56.7
DATAMODELS†	$84.1_{9.3}$	58.9	$60.6_{3.3}$	54.3
AME-ICL	**88.7**$_{3.4}$	**76.5**	**66.5**$_{2.0}$	**61.1**

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 2, 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. [36]. 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 15.1% and 10.7%, respectively, on average. Moreover, compared to the best performance in the unlabeled scenario, our method outperforms by 4.9%. Other baselines, such as UN-TOPPROMPTS and UN-CONDACC, perform better than UN-Vanilla ICL but notably worse than our approach.

5.2 Imbalanced Labels in Prompts

Previous studies [34, 67] have revealed that imbalanced class distributions in demonstrations significantly impair the performance of ICL. Specifically, models tend to favor the majority class in their outputs. 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 [35] and Channel ICL [34]. 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. Figs. 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. It becomes evident that the performance of Vanilla ICL is 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.

Table 5: Accuracy comparison between Vanilla ICL and AME-ICL on the SST-2 and Subj datasets, employing seven large PLMs. All models utilize valuable demonstration examples selected by the GPT-2-0.8B model.

	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	**87.5**$_{3.3}$	**80.8**	**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	**88.7**$_{2.9}$	**80.0**	**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	**90.0**$_{4.3}$	**81.7**	**82.4**$_{3.0}$	**74.0**
GPTJ-6B				
Vanilla ICL	$77.8_{11.2}$	50.8	$59.8_{8.3}$	50.1
AME-ICL	**88.3**$_{3.1}$	**80.2**	**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	**89.7**$_{3.5}$	**80.2**	**77.6**$_{4.1}$	**72.7**
OPT-13B				
Vanilla ICL	$68.5_{14.0}$	50.0	$60.9_{10.2}$	49.8
AME-ICL	**90.6**$_{3.5}$	**82.1**	**78.3**$_{2.8}$	**72.5**
LLaMA-33B				
Vanilla ICL	$93.6_{7.2}$	71.4	$83.1_{8.0}$	66.8
AME-ICL	**96.8**$_{2.7}$	**88.6**	**89.9**$_{3.0}$	**82.5**

5.3 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 Eq. (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 3, **AME-ICL persistently achieves the**

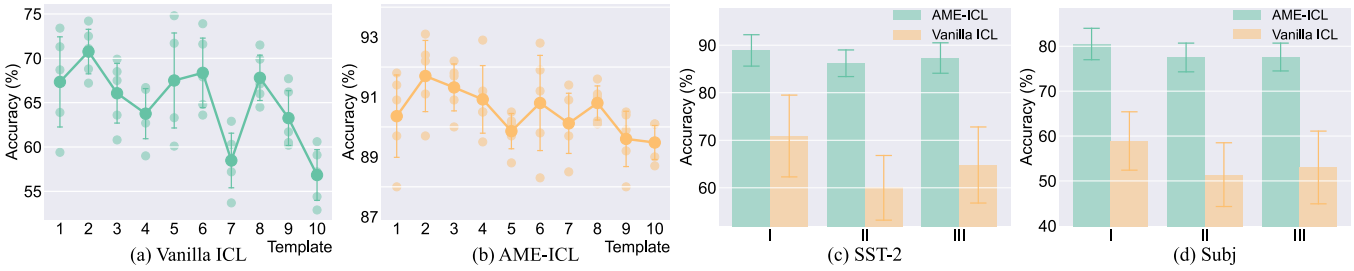

Figure 4: (a) and (b): Accuracy comparison between Vanilla ICL and AME-ICL on the SST-2 dataset using the OPT-13B model across ten templates. (c) and (d): Accuracy comparison under three different permutation settings on the SST-2 and Subj datasets utilizing the GPT-2-1.5B model.

Table 6: The four generalized valuable examples calculated by AME-ICL shared among the seven PLMs in the SST-2 dataset. All seven models exhibit high average accuracy and low variance when utilizing these examples as demonstrations.

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

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 2, 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.

5.4 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 [3]. 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 [33] and BoolQ Contrast Set [11] as our target tasks, respectively. The findings presented in Table 4 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**.

5.5 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 six other large PLMs (i.e., GPT-2-1.5B, GPT-Neo-2.7B, GPTJ-6B, OPT-6.7B, OPT-13B, and LLaMA-33B). Two datasets, SST-2 and Subj, are employed for this purpose. The experimental findings, as reported in Table 5, reveal that the performance of various PLMs utilizing valuable samples selected by the GPT-2-0.8B model consistently surpasses that of Vanilla ICL. This suggests that **the valuable training samples demonstrate transferability across various PLMs**. Moreover, our findings suggest the presence of successful factors among the valuable training examples. We present four generalized samples in Table 6 and encourage future research to investigate the distinguishing characteristics of these valuable examples. Additionally, our proposed AME-ICL method performs well even on gigantic PLMs, such as LLaMA-33B.

5.6 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 Figs. 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 demonstration examples selected by our proposed AME-ICL method encapsulate richer and more valuable knowledge of the task.

5.7 Varying Templates

Previous studies have highlighted that the ICL performance is sensitive to the applied templates for demonstration examples [53, 67].

Table 7: The templates utilized for examining the influence of formats on the ICL performance. An example from the training set of the SST-2 dataset is provided for illustration purposes.

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

To assess the performance of AME-ICL across different templates, we apply ten templates on the SST-2 dataset, as presented in Table 7, following those outlined by Zhao et al. [67]. The OPT-13B model is utilized for this purpose. The accuracy of Vanilla ICL and AME-ICL across these ten templates is depicted in Figs. 4(a) and (b). It is observed that certain templates yield higher average performance than others. Nonetheless, **AME-ICL consistently enhances accuracy compared to Vanilla ICL, all the while decreasing performance variance across diverse templates**.

5.8 Varying Permutations of Demonstrations

Prior studies have highlighted that the effectiveness of ICL can be affected by the permutation of demonstration examples [30, 67]. 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 Figs. 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.

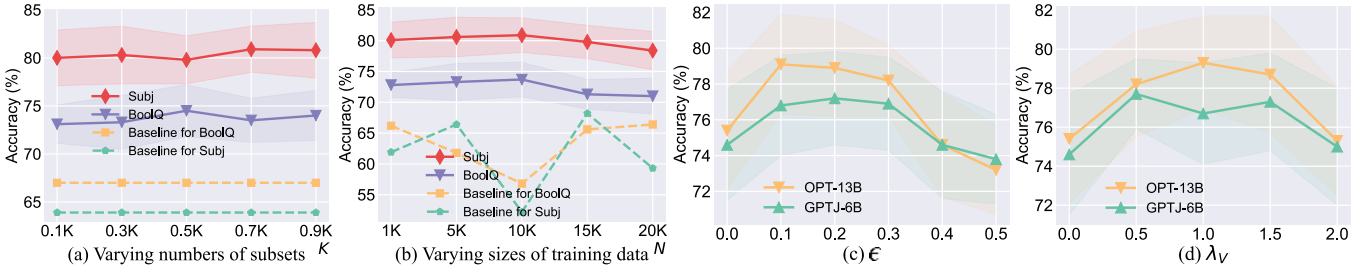

Figure 5: (a) and (b): Accuracy of the BoolQ and Subj datasets with varying numbers of training subsets (K) and varying sizes of training data (N) on the OPT-13B model, with $\mathcal{M}$ setting to ten. (c) and (d): Sensitivity analysis for the perturbation bound ϵ and the modulating factor λ_V, using the GPTJ-6B and OPT-13B models. The average accuracy across all tasks is reported.

Table 8: Results of ablation studies for different configurations of data values on the Subj and BoolQ datasets employing the GPTJ-6B model.

Values		Subj		BoolQ	
V^g	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**

5.9 Efficiency and Scalability

We explore the efficiency of AME-ICL to emphasize its scalability. The additional time consumption of AME-ICL primarily arises from utilizing prompts generated from diverse training subsets for inference. As depicted in Fig. 5(a), with a fixed training size of 1,000, satisfactory results can be attained with just 0.1K 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. 5(b), when the size of the training data becomes large (i.e., 15K and 20K), there is only a slight decline in ICL performance, which still significantly outperforms the baseline. Consequently, **AME-ICL significantly enhances efficiency compared to baseline methods such as CONDACC and DATAMODELS**, which consume hundreds of GPU hours due to the need for inference with numerous prompts (exceeding 50K prompts in the case of a training set size of 1,000). Fig. 5(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.

5.10 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 (V^g), only the robustness value (V^r), and both values combined. The results, as reported in Table 8, 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 V^g generally yields superior results compared to solely focusing on V^r.

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 Figs. 5(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.

6 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. Firstly, 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. Thirdly, 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] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. https://api.semanticscholar.org/CorpusID:245758737

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models Are Few-shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 1877–1901.

[3] Ting-Yun Chang and Robin Jia. 2023. Data Curation Alone Can Stabilize In-Context Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 8123–8144.

[4] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 2924–2936.

[5] Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 3586–3596.

[6] Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay-Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-Based Reasoning for Natural Language Queries over Knowledge Bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 9594–9611.

[7] Laurent Davezies, Xavier D'Haultfoeuille, and Louise Laage. 2021. Identification and Estimation of Average Marginal Effects in Fixed Effects Logit Models. *arXiv preprint arXiv:2105.00879* (2021).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[9] Naoki Egami and Kosuke Imai. 2019. Causal Interaction in Factorial Experiments: Application to Conjoint Analysis. *J. Amer. Statist. Assoc.* 114, 526 (2019), 529–540.

[10] Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating Label Biases for In-Context Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 14014–14031.

[11] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 1307–1323.

[12] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the International Conference on Machine Learning*. 2242–2251.

[13] Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical Calibration for Few-shot Learning of Language Models. In *Proceedings of the International Conference on Learning Representations*.

[14] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface Form Competition: Why the Highest Probability Answer Isn't Always Right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 7038–7051.

[15] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Predicting Predictions from Training Data. In *Proceedings of the 39th International Conference on Machine Learning*. 9525–9587.

[16] Kevin Fu Jiang, Weixin Liang, James Zou, and Yongchan Kwon. 2023. OpenDataVal: A Unified Benchmark for Data Valuation. In *Proceedings of the 37th Conference on Neural Information Processing Systems*. 28624–28647.

[17] Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. 2023. LAVA: Data Valuation without Pre-specified Learning Algorithms. In *Proceedings of the 11th International Conference on Learning Representations*.

[18] Pang Wei Koh and Percy Liang. 2017. Understanding Black-Box Predictions via Influence Functions. In *Proceedings of the International Conference on Machine Learning*. 1885–1894.

[19] Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. In-Context Learning Learns Label Relationships but Is Not Conventional Learning. In *Proceedings of the 12th International Conference on Learning Representations*.

[20] Yongchan Kwon and James Zou. 2021. Beta Shapley: A Unified and Noise-Reduced Data Valuation Framework for Machine Learning. *arXiv preprint arXiv:2110.14049* (2021).

[21] Yongchan Kwon and James Zou. 2023. Data-OOB: Out-of-bag Estimate as a Simple and Efficient Data Value. In *Proceedings of the 40th International Conference on Machine Learning*. 18135–18152.

[22] Guillaume Lecué and Shahar Mendelson. 2018. Regularization and the Small-Ball Method I: Sparse Recovery. *The Annals of Statistics* 46, 2 (2018), 611–641.

[23] Thomas J Leeper. 2017. Interpreting Regression Results Using Average Marginal Effects with R's Margins. *The comprehensive R Archive Network* 32 (2017), 1–32.

[24] Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2022. The Inductive Bias of In-Context Learning: Rethinking Pretraining Example Design. In *Proceedings of the International Conference on Learning Representations*.

[25] Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse Demonstrations Improve In-Context Compositional Generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 1401–1422.

[26] Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. 2023. In-Context Learning with Many Demonstration Examples. *arXiv preprint arXiv:2302.04931* (2023).

[27] Xiaonan Li and Xipeng Qiu. 2023. Finding Support Examples for In-Context Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 6219–6235.

[28] Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. 2022. Measuring the Effect of Training Data on Deep Learning Predictions via Randomized Experiments. In *Proceedings of the International Conference on Machine Learning*. 13468–13504.

[29] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of the 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. 100–114.

[30] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-shot Prompt Order Sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 8086–8098.

[31] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the Advances in Neural Information Processing Systems*. 4768–4777.

[32] Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games. *Proceedings of the ACM on Management of Data* 2, 1 (2024), 1–28.

[33] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. 142–150.

[34] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy Channel Language Model Prompting for Few-shot Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 5316–5330.

[35] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to Learn in Context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*. 2791–2809.

[36] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11048–11064.

[37] Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. 271–278.

[38] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep Learning on a Data Diet: Finding Important Examples Early in Training. In *Proceedings of the Advances in Neural Information Processing Systems*. 20596–20607.

[39] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True Few-shot Learning with Language Models. In *Proceedings of the Advances in Neural Information Processing Systems*. 11054–11070.

[40] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2463–2473.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. In *OpenAI Blog*. 1–12.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[43] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 3982–3992.

[44] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A Survey of Deep Active Learning. *Comput. Surveys* 54, 9 (2021), 1–40.

[45] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to Retrieve Prompts for In-Context Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*. 2655–2671.
[46] Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. 2021. A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks. In *Proceedings of the International Conference on Learning Representations*.
[47] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *Proceedings of the 2018 International Conference on Learning Representations*.
[48] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing Data Science through Interactive Curation of ML Pipelines. In *Proceedings of the 2019 International Conference on Management of Data*. 1171–1188.
[49] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. XRICL: Cross-Lingual Retrieval-Augmented In-Context Learning for Cross-Lingual Text-to-SQL Semantic Parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5248–5259.
[50] Richard Shin, Christopher H Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained Language Models Yield Few-shot Semantic Parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 7699–7715.
[51] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1631–1642.
[52] Tianhong Song. 2015. Provenance-Driven Data Curation Workflow Analysis. In *Proceedings of the 2015 ACM SIGMOD on PhD Symposium*. 45–50.
[53] Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An Information-Theoretic Approach to Prompt Engineering without Ground Truth Labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 819–862.
[54] Mihail Stoian. 2023. Fast Joint Shapley Values. In *Proceedings of the Companion of the 2023 International Conference on Management of Data*. 285–287.
[55] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Selective Annotation Makes Language Models Better Few-shot Learners. In *Proceedings of the International Conference on Learning Representations*.
[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).
[57] Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.
[58] Jiachen T. Wang and Ruoxi Jia. 2023. Data Banzhaf: A Robust Data Valuation Framework for Machine Learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. 6388–6421.
[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proceedings of the Advances in Neural Information Processing Systems*. 24824–24837.
[60] Brian Williamson and Jean Feng. 2020. Efficient Nonparametric Statistical Inference on Population Feature Importance Using Shapley Values. In *Proceedings of the International Conference on Machine Learning*. 10282–10291.
[61] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-Adaptive In-Context Learning: An Information Compression Perspective for In-Context Example Selection and Ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 1423–1436.
[62] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-Context Learning as Implicit Bayesian Inference. In *Proceedings of the International Conference on Learning Representations*.
[63] Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data Valuation Using Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning*. 10842–10851.
[64] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. OPT: Open Pre-trained Transformer Language Models. *arXiv preprint arXiv:2205.01068* (2022).
[65] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-Level Convolutional Networks for Text Classification. In *Proceedings of the Advances in Neural Information Processing Systems*. 649–657.
[66] Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active Example Selection for In-Context Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 9134–9148.
[67] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the International Conference on Machine Learning*. 12697–12706.
[68] Hui Zou and Trevor Hastie. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67, 2 (2005), 301–320.