# Enhancing In-Context Learning via Implicit Demonstration Augmentation

**Xiaoling Zhou**[1], **Wei Ye**[1*], **Yidong Wang**[1], **Chaoya Jiang**[1], **Zhemg Lee**[2], **Rui Xie**[1], **Shikun Zhang**[1*]
[1]National Engineering Research Center for Software Engineering, Peking University, China
[2]Tianjin University, Tianjin, China
xiaolingzhou@stu.pku.edu.cn, {wye,zhangsk}@pku.edu.cn

## Abstract

The emergence of in-context learning (ICL) enables large pre-trained language models (PLMs) to make predictions for unseen inputs without updating parameters. Despite its potential, ICL's effectiveness heavily relies on the quality, quantity, and permutation of demonstrations, commonly leading to suboptimal and unstable performance. In this paper, we tackle this challenge for the first time from the perspective of demonstration augmentation. Specifically, we start with enriching representations of demonstrations by leveraging their deep feature distribution. We then theoretically reveal that when the number of augmented copies approaches infinity, the augmentation is approximately equal to a novel logit calibration mechanism integrated with specific statistical properties. This insight results in a simple yet highly efficient method that significantly improves the average and worst-case accuracy across diverse PLMs and tasks. Moreover, our method effectively reduces performance variance among varying demonstrations, permutations, and templates, and displays the capability to address imbalanced class distributions.

## 1 Introduction

Large pre-trained language models (PLMs) have showcased exceptional abilities in in-context learning (ICL) (Brown et al., 2020; Wang et al., 2023; Rubin et al., 2022), which assists the model in discerning the underlying patterns within demonstrations and make more accurate predictions (Chan et al., 2022; Wu et al., 2023). As a new paradigm, ICL offers compelling advantages, allowing for natural language interaction with PLMs (Wei et al., 2022; Yang et al., 2023), as well as reduced computational costs (Li et al., 2023a; Rubin et al., 2022).

While promising, ICL's performance is highly dependent on provided demonstrations and templates (Liu et al., 2022; Zhang et al., 2022b; Sorensen et al., 2022), resulting in subpar and unstable performance. This promotes research aimed at improving the quality (Rubin et al., 2022; Li et al., 2023b), quantity (Li et al., 2023a; Choi et al., 2022), and permutations (Lu et al., 2022; Tang et al., 2023) of demonstrations. Other research avenues include prediction adjustment (Zhao et al., 2021; Han et al., 2023; Fei et al., 2023) and learning process design (e.g., channel models (Min et al., 2022a) and meta-training frameworks (Min et al., 2022b)). Despite ongoing efforts, ICL still struggles with efficiently and reliably capturing sufficient knowledge from context, leaving performance stability as a persistent bottleneck.

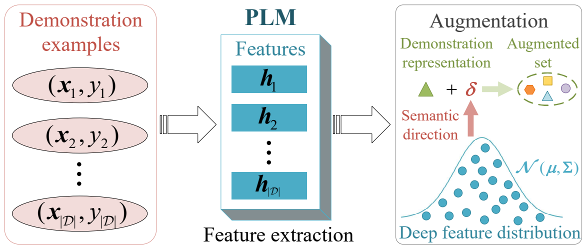



Figure 1: Illustration for demonstration augmentation using semantic directions (vectors) sampled from the deep feature distribution of demonstration examples.

In this study, we propose enriching contextual knowledge for PLMs by augmenting demonstrations. We first attempt to enhance the representation of demonstrations by transforming them along semantic directions sampled from the deep feature space of demonstration examples, as depicted in Figure 1. This operation stems from the observation that the deep features in a network are usually linearized (Bengio et al., 2013; Cheung and Yeung, 2021; Cho, 2016), implying the existence of numerous semantic directions within the deep feature space, hence potentially enabling us to incorporate richer contextual knowledge without extending input length. From this novel perspective, we theoretically prove that when the number of augmented

*Corresponding authors.



*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2810–2828
August 11-16, 2024 ©2024 Association for Computational Linguistics

pieces approaches infinity, its effect approximately equals a logit adjustment operation. Specifically, we derive a refined Softmax function that integrates the statistical properties of demonstrations. Consequently, rather than explicitly executing the augmentation procedure, we can efficiently conduct implicit demonstration augmentation using the derived prediction function, obtaining an improved ICL method with theoretical guidance.

We conduct extensive experiments across seven PLMs and various classification tasks. The empirical results demonstrate that our approach remarkably enhances prediction accuracy and reduces performance variability across different demonstrations, permutations, and templates. Notably, our method is straightforward, effective, and generalizable, enabling seamless integration with other ICL methods to enhance their performance.

Our contributions can be summarized as follows:

- We introduce **I**mplicit **D**emonstration **A**ugmentation-based **ICL** (IDAICL), a pioneering work that incorporates demonstration augmentation into ICL. Instead of solely enhancing demonstration quality, quantity, or order, our method explores context augmentation within the deep feature space, offering a new perspective to enrich demonstrations bypassing input length limitations.
- We theoretically establish that as the number of augmented pieces approaches infinity, our augmentation strategy approximates a logit-adjusted prediction function that integrates statistical properties derived from the input data distribution. Equipped with this function, IDAICL provides a straightforward yet theory-guided solution to enhance ICL.
- Extensive experiments conducted across diverse tasks and PLMs conclusively illustrate that IDAICL considerably improves average and worst-case accuracy compared to existing ICL methods. Moreover, it effectively enhances performance stability.

## 2 Background and Related Work

### 2.1 In-Context Learning

Brown et al. (2020) showcased the ICL capability of PLMs, wherein PLMs generate predictions solely based on a concatenation of training examples for few-shot learning without updating parameters. Subsequent studies (Holtzman et al., 2021; Min et al., 2022a,b) have developed this approach, yielding promising outcomes across various tasks. Nevertheless, recent research has uncovered certain limitations. To begin with, the volume of input knowledge for each query is constrained by the maximum input length of PLMs (Hao et al., 2022), and the computational cost increases as the number of demonstrations grows (Li et al., 2023a), making it challenging to integrate significant knowledge from demonstrations to PLMs. Additionally, ICL's performance is sensitive to the input of PLMs (Davison et al., 2019; Jiang et al., 2020), thus exhibiting high variance and poor worst-case accuracy (Perez et al., 2021; Lu et al., 2022).

Researchers have explored various techniques to address the biases and instability of ICL. These techniques encompass learning process design (Min et al., 2022a,b), demonstration retrieval (Rubin et al., 2022; Zhang et al., 2022b), prompt engineering (Sorensen et al., 2022; Lu et al., 2022), and prediction calibration (Zhao et al., 2021; Fei et al., 2023). However, these methods have yet to fully address the issue of severely limited knowledge transfer from demonstrations to large PLMs.

### 2.2 Data Augmentation

Data augmentation (Chen et al., 2023), which involves artificially creating training data through transformations, is a well-established research area in machine learning. Although data augmentation techniques have undergone extensive exploration in diverse machine learning domains (Maharana et al., 2022; Shorten and Khoshgoftaar, 2019), applying them to text data poses challenges due to the complexity of preserving labels during textual transformations (Kobayashi, 2018). Nonetheless, data augmentations in the latent space, such as adversarial training (Zhang et al., 2022a; Zhu et al., 2020; Cheng et al., 2020), interpolation (Chen et al., 2022b; Wu et al., 2022), and generative techniques (Li et al., 2022; Malandrakis et al., 2019), have demonstrated notable enhancements when applied alongside large PLMs.

Recently, Wang et al. (2019) introduced the concept of implicit data augmentation in the context of image classification. This approach involves transforming training data within the deep feature space and boils down to the optimization of a novel robust loss function. Subsequent studies (Chen et al., 2022c; Li et al., 2021; Zhou and Wu, 2023a) for image classification tasks have further improved

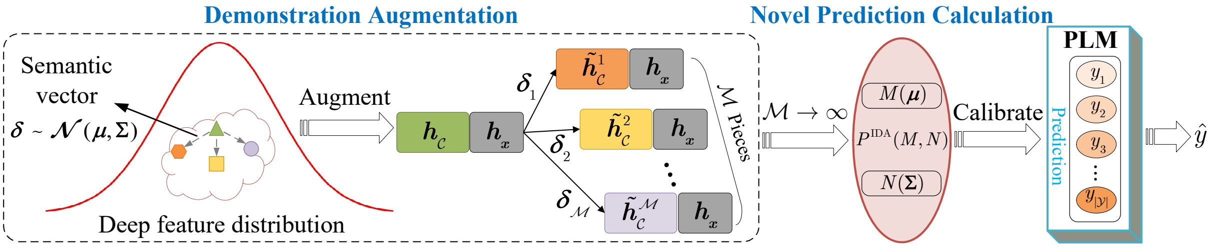



Figure 2: An overview of IDAICL: For each contextual input, our goal is to augment the deep feature of demonstrations for $\mathcal{M}$ pieces, using semantic vectors $\boldsymbol{\delta}$ drawn from the deep feature distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of demonstration examples linked to all queries. When $\mathcal{M}$ approaches infinity, we derive a novel prediction function, which incorporates two modulating factors: $M(\boldsymbol{\mu})$ and $N(\boldsymbol{\Sigma})$, to calibrate the original predictions.

upon this approach. This study introduces an algorithm for implicitly augmenting demonstrations within the realm of ICL.

## 3 Methodology

### 3.1 In-Context Learning with PLMs

Considering a PLM $\mathcal{G}$, this study focuses on the following task: given a query input text $\boldsymbol{x}$ and a candidate answer set $\mathcal{Y} = \{y_1, y_2, \cdots, y_{|\mathcal{Y}|}\}$, we aim to predict the answer $\hat{y}$ based on $m$ demonstration examples $\mathcal{C} = \{c_1, c_2, \cdots, c_m\}$, where each $c_i$ represents a training example $(\boldsymbol{x}_i, y_i)$ after template formulation and $m$ denotes the quantity of demonstration examples for each test sample. Formally, give a model $\mathcal{G}$, we first compute the probability of each answer $y_j$:

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

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

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

To simplify, the contextual input is denoted as $\tilde{\boldsymbol{x}} = [\mathcal{C}, \boldsymbol{x}]$ in the subsequent text. Then, the probability of answer $y_j$, represented as $P_{\mathcal{G}}(y_j|\tilde{\boldsymbol{x}})$, is computed using the Softmax function[1]:

$$P_{\mathcal{G}}(y_j|\tilde{\boldsymbol{x}}) := P_{\mathcal{G}}(y_j|\boldsymbol{h}_{\tilde{\boldsymbol{x}}}) = \frac{e^{\boldsymbol{w}_{y_j}^T \boldsymbol{h}_{\tilde{\boldsymbol{x}}} + b_{y_j}}}{\sum_k e^{\boldsymbol{w}_k^T \boldsymbol{h}_{\tilde{\boldsymbol{x}}} + b_k}}, \tag{3}$$

where $\boldsymbol{h}_{\tilde{\boldsymbol{x}}} = \mathcal{G}(\tilde{\boldsymbol{x}})$ signifies the hidden state of the last block at the final position for $\tilde{\boldsymbol{x}}$. $\boldsymbol{w}_k$ and $b_k$ are the weight vector and bias corresponding to the final fully connected layer for the $k$-th token.

[1]We begin by examining situations in which the answer comprises a single token, and our subsequent analysis is equally applicable to scenarios involving multiple tokens.

### 3.2 Demonstration Augmentation

Recognizing the established efficacy of data augmentation in machine learning (Feng et al., 2021), this study investigates demonstration augmentation and suggests enhancing the deep features of demonstrations by transforming them along semantic directions sampled from the deep feature space of demonstration examples. This strategy is motivated by the intriguing observation that the deep features in networks are often linearized (Bengio et al., 2013; Chen et al., 2022a). Building on this observation, we hypothesize that $\boldsymbol{h}_{\tilde{\boldsymbol{x}}}$ lies within the subspace spanned by $\boldsymbol{h}_{\mathcal{C}}$ and $\boldsymbol{h}_{\boldsymbol{x}}$: $\boldsymbol{h}_{\tilde{\boldsymbol{x}}} = \alpha \boldsymbol{h}_{\mathcal{C}} + \beta \boldsymbol{h}_{\boldsymbol{x}}$, where $\boldsymbol{h}_{\mathcal{C}}$ and $\boldsymbol{h}_{\boldsymbol{x}}$ represent the components of $\boldsymbol{h}_{\tilde{\boldsymbol{x}}}$ linked respectively to the demonstrations and the query. The necessity of this assumption stems from intricate relationships among token representations and the exclusive augmentation of the component related to demonstrations. Notably, this decomposition is not necessary in practical applications. In the subsequent text, we directly refer to $\alpha \boldsymbol{h}_{\mathcal{C}}$ and $\beta \boldsymbol{h}_{\boldsymbol{x}}$ as $\boldsymbol{h}_{\mathcal{C}}$ and $\boldsymbol{h}_{\boldsymbol{x}}$.

To augment $\boldsymbol{h}_{\mathcal{C}}$, we randomly sample vectors from the deep feature space of demonstrations. In particular, vectors are drawn from a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the feature mean and covariance matrix. These statistical properties are estimated from the deep features of the demonstration set $\mathcal{D}$, which includes demonstration examples linked to all queries. The feature mean $\boldsymbol{\mu}$ is computed as

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{D}|} \sum\nolimits_{i=1}^{|\mathcal{D}|} \boldsymbol{h}_i, \tag{4}$$

where $\boldsymbol{h}_i = \mathcal{G}(c_i)$ represents the hidden state of the last block at the final position for the $i$-th demonstration example $c_i$ in $\mathcal{D}$, and $|\mathcal{D}|$ denotes the size of $\mathcal{D}$. The covariance matrix $\boldsymbol{\Sigma}$ is computed as

$$\boldsymbol{\Sigma} = \frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}(\boldsymbol{h}_i - \boldsymbol{\mu})^T(\boldsymbol{h}_i - \boldsymbol{\mu}). \quad (5)$$

Subsequently, $\boldsymbol{h}_{\mathcal{C}}$ is shifted in the extracted semantic vectors, resulting in augmented features, $\tilde{\boldsymbol{h}}_{\mathcal{C}}$, which follows

$$\tilde{\boldsymbol{h}}_{\mathcal{C}} \sim \mathcal{N}\left(\boldsymbol{h}_{\mathcal{C}} + \lambda\boldsymbol{\mu}, \lambda\boldsymbol{\Sigma}\right), \quad (6)$$

where $\lambda$ refers to a positive coefficient controlling the strength of semantic augmentation. In real-world applications, it can be directly assigned a value of 0.5. Sensitivity tests for $\lambda$ are discussed in Section 5.4.

### 3.3 Novel Prediction Function

Selecting the answer with the highest probability is equivalent to favoring the answer with the lowest inverse probability. Therefore, the prediction can be determined by

$$\hat{y} = \arg\min_{y_j \in \mathcal{Y}} P_{\mathcal{G}}\left(y_j \mid \boldsymbol{h}_{\tilde{\boldsymbol{x}}}\right)^{-1}. \quad (7)$$

Assume that each $\boldsymbol{h}_{\mathcal{C}}$ is augmented for $\mathcal{M}$ times, resulting in an augmented demonstration feature set $\{\tilde{\boldsymbol{h}}_{\mathcal{C}}^1, \cdots, \tilde{\boldsymbol{h}}_{\mathcal{C}}^{\mathcal{M}}\}$ with size $\mathcal{M}$. Here, $\tilde{\boldsymbol{h}}_{\mathcal{C}}^i$ represents the $i$-th augmented feature for $\boldsymbol{h}_{\mathcal{C}}$. Then, the final prediction for the query $\boldsymbol{x}$ depends on all augmented features of $\boldsymbol{h}_{\mathcal{C}}$ and can be expressed as

$$P_{y_j}^{\mathcal{M}}(\tilde{\boldsymbol{x}}) = \frac{1}{\mathcal{M}}\sum_{i=1}^{\mathcal{M}} P_{\mathcal{G}}(y_j|\tilde{\boldsymbol{h}}_{\mathcal{C}}^i, \boldsymbol{h}_{\boldsymbol{x}})^{-1}, \quad (8)$$

$$\hat{y} = \arg\min_{y_j \in \mathcal{Y}} P_{y_j}^{\mathcal{M}}(\tilde{\boldsymbol{x}}). \quad (9)$$

Given that the performance of ICL benefits from an increased number of demonstration instances (Liu et al., 2022; Wu et al., 2023), we explore the scenario of augmenting an infinite number of times for the deep representation of demonstrations. Subsequently, an easily computable surrogate for the expected prediction can be derived, resulting in a highly efficient implementation. The whole pipeline of IDAICL is depicted in Figure 2.

As $\mathcal{M} \to \infty$, on the basis of the aforementioned decomposition of $\boldsymbol{h}_{\tilde{\boldsymbol{x}}}$, the expected prediction for answer $y_j$ (denoted as $P_{y_j}^{\infty}$) within the augmented feature set can be expressed as follows:

$$P_{y_j}^{\infty}(\tilde{\boldsymbol{x}}) = \mathbb{E}_{\tilde{\boldsymbol{h}}_{\mathcal{C}}}[\sum_k e^{\Delta\boldsymbol{w}_{k,y_j}^T(\tilde{\boldsymbol{h}}_{\mathcal{C}}+\boldsymbol{h}_{\boldsymbol{x}})+\Delta b_{k,y_j}}], \quad (10)$$

where $\Delta\boldsymbol{w}_{k,y_j} = \boldsymbol{w}_k - \boldsymbol{w}_{y_j}$ and $\Delta b_{k,y_j} = b_k - b_{y_j}$.

However, accurately calculating $P_{y_j}^{\infty}$ is challenging. Alternatively, we proceed to derive a surrogate calculation for it. Applying the linearity of expectation, Eq. (10) can be expressed as:

$$P_{y_j}^{\infty}(\tilde{\boldsymbol{x}}) = \sum_k \mathrm{E}_{\tilde{\boldsymbol{h}}_{\mathcal{C}}}[e^{\Delta\boldsymbol{w}_{k,y_j}^T(\tilde{\boldsymbol{h}}_{\mathcal{C}}+\boldsymbol{h}_{\boldsymbol{x}})+\Delta b_{k,y_j}}]. \quad (11)$$

Given that $\tilde{\boldsymbol{h}}_{\mathcal{C}}$ is a Gaussian random variable conforming to $\mathcal{N}\left(\boldsymbol{h}_{\mathcal{C}} + \lambda\boldsymbol{\mu}, \lambda\boldsymbol{\Sigma}\right)$, we know that $\Delta\boldsymbol{w}_{k,y_j}^T\tilde{\boldsymbol{h}}_{\mathcal{C}}$ follows the multivariate normal distribution: $\mathcal{N}(\Delta\boldsymbol{w}_{k,y_j}^T\left(\boldsymbol{h}_{\mathcal{C}} + \lambda\boldsymbol{\mu}\right), \lambda\Delta\boldsymbol{w}_{k,y_j}^T\boldsymbol{\Sigma}\Delta\boldsymbol{w}_{k,y_j})$. Then, utilizing the moment-generating function

$$\mathbb{E}[e^{tX}] = e^{t\mu + \frac{1}{2}t^2\sigma^2}, X \sim \mathcal{N}(\mu, \sigma^2), \quad (12)$$

Eq. (11) can be derived as

$$P_{y_j}^{\infty}(\tilde{\boldsymbol{x}}) = \sum_k M_{k,y_j} N_{k,y_j} e^{\Delta\boldsymbol{w}_{k,y_j}^T(\boldsymbol{h}_{\mathcal{C}}+\boldsymbol{h}_{\boldsymbol{x}})+\Delta b_{k,y_j}}, \quad (13)$$

where $M_{k,y_j} = \exp(\lambda\Delta\boldsymbol{w}_{k,y_j}^T\boldsymbol{\mu})$ and $N_{k,y_j} = \exp(\frac{\lambda}{2}\Delta\boldsymbol{w}_{k,y_j}^T\boldsymbol{\Sigma}\Delta\boldsymbol{w}_{k,y_j})$.

Subsequently, our newly proposed prediction function, referred to as IDA-Softmax, is defined as

$$P_{y_j}^{\text{IDA}}(\tilde{\boldsymbol{x}}) := \sum_k M_{k,y_j} N_{k,y_j} e^{\Delta\boldsymbol{w}_{k,y_j}^T\boldsymbol{h}_{\tilde{\boldsymbol{x}}}+\Delta b_{k,y_j}}. \quad (14)$$

Consequently, instead of conducting the augmentation process explicitly, we can directly employ IDA-Softmax, $P_{y_j}^{\text{IDA}}$, for prediction. IDA-Softmax essentially utilizes two modulating factors associated with statistical properties derived from $\mathcal{D}$ to calibrate the sample logits. Previous studies (Min et al., 2022c; Chan et al., 2022) have underscored the pivotal role of knowledge about the input data distribution in predictions made by PLMs. Intuitively, PLMs can better capture the patterns and underlying structures within data, such as the spatial relationships between demonstrations and queries, ultimately enhancing their prediction performance.

Furthermore, to mitigate the imbalance among different answer types in demonstrations (Holtzman et al., 2021; Zhao et al., 2021), we adopt a post-hoc adjustment approach inspired by Menon et al. (2021), which adjusts predictions by considering the class proportions within $\mathcal{D}$. Thus, the prediction for answer $y_j$ is computed as

$$\tilde{P}_{y_j}^{\text{IDA}}(\tilde{\boldsymbol{x}}) = P_{y_j}^{\text{IDA}}(\tilde{\boldsymbol{x}}) + \tau \log \pi_{y_j}, \quad (15)$$

where $\tau$ is a positive hyperparameter, and $\pi_{y_j}$ demotes the proportion of answer $y_j$ in $\mathcal{D}$. In practical applications, the value of $\tau$ can be fixed at 1.

| PLM | Method | m | SST-2 | SST-5 | MR | CR | Amazon | Subj | TREC | DBPedia | AGNews | CB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPT-2 0.8B | Vanilla ICL | 4 | $57.6_{7.1}$ | $30.4_{6.3}$ | $59.3_{6.5}$ | $56.8_{8.4}$ | $32.7_{8.5}$ | $57.6_{5.4}$ | $34.9_{10.3}$ | $40.5_{7.2}$ | $44.5_{7.9}$ | $35.1_{9.3}$ |
| | IDAICL | | $86.4_{1.4}$ | $38.3_{2.9}$ | $82.2_{2.3}$ | $78.4_{0.7}$ | $46.7_{3.5}$ | $77.0_{2.3}$ | $47.5_{2.0}$ | $81.3_{1.8}$ | $73.9_{2.4}$ | $41.5_{2.0}$ |
| | Vanilla ICL | 8 | $69.7_{9.0}$ | $32.4_{8.6}$ | $63.9_{7.7}$ | $60.8_{8.1}$ | $34.1_{6.2}$ | $59.7_{8.7}$ | $40.4_{6.3}$ | $62.6_{13.6}$ | $49.2_{8.4}$ | $38.8_{7.6}$ |
| | IDAICL | | $88.0_{2.3}$ | $39.6_{1.9}$ | $84.9_{2.4}$ | $85.6_{2.5}$ | $47.9_{2.6}$ | $79.9_{0.8}$ | $50.3_{3.3}$ | $86.5_{2.9}$ | $76.8_{1.7}$ | $43.3_{3.4}$ |
| | Vanilla ICL | 12 | $74.7_{8.3}$ | $33.7_{7.6}$ | $64.4_{9.4}$ | $68.7_{9.7}$ | $36.0_{6.6}$ | $60.7_{7.7}$ | $40.5_{7.8}$ | $64.5_{5.4}$ | $51.1_{8.0}$ | $40.4_{8.5}$ |
| | IDAICL | | $88.5_{2.1}$ | $40.1_{2.7}$ | $85.2_{3.1}$ | $86.8_{1.4}$ | $49.6_{2.2}$ | $80.4_{2.1}$ | $51.4_{1.6}$ | $87.3_{2.7}$ | $77.9_{2.0}$ | $44.6_{2.2}$ |
| | MetaICL | 12 | $80.8_{6.2}$ | $35.8_{4.7}$ | $75.3_{5.6}$ | $77.6_{8.1}$ | $48.9_{6.7}$ | $73.5_{8.8}$ | $48.6_{6.1}$ | $80.4_{7.8}$ | $66.8_{0.7}$ | $43.1_{4.1}$ |
| | +IDAICL | | $89.3_{1.7}$ | $\underline{42.6}_{2.4}$ | $85.8_{1.7}$ | $87.9_{1.5}$ | $\underline{51.7}_{0.7}$ | $\underline{82.6}_{2.4}$ | $53.7_{2.5}$ | $\underline{89.4}_{4.1}$ | $78.3_{1.1}$ | $\mathbf{47.9}_{2.8}$ |
| | Channel ICL | 12 | $85.2_{3.6}$ | $38.4_{4.3}$ | $80.8_{4.7}$ | $82.0_{4.6}$ | $43.6_{5.1}$ | $69.8_{9.8}$ | $44.1_{8.7}$ | $77.6_{12.9}$ | $69.5_{6.7}$ | $42.4_{5.2}$ |
| | +IDAICL | | $\mathbf{90.5}_{2.3}$ | $41.8_{2.7}$ | $\mathbf{87.7}_{1.6}$ | $\mathbf{89.5}_{1.2}$ | $50.8_{2.4}$ | $80.5_{0.9}$ | $52.9_{1.6}$ | $87.8_{2.4}$ | $\underline{81.0}_{2.5}$ | $46.3_{3.3}$ |
| | EPR | 12 | $81.9_{2.1}$ | $39.9_{1.8}$ | $78.1_{2.4}$ | $80.6_{0.6}$ | $49.1_{2.4}$ | $80.1_{2.2}$ | $\underline{76.2}_{1.1}$ | $87.1_{1.0}$ | $80.9_{0.8}$ | $44.8_{2.3}$ |
| | +IDAICL | | $\underline{90.1}_{1.1}$ | $\mathbf{43.9}_{1.2}$ | $\underline{86.4}_{2.0}$ | $\underline{88.6}_{0.6}$ | $\mathbf{52.5}_{1.7}$ | $\mathbf{83.6}_{1.0}$ | $\mathbf{79.1}_{0.9}$ | $\mathbf{90.8}_{0.7}$ | $\mathbf{83.7}_{0.5}$ | $\underline{46.7}_{2.1}$ |
| GPT-2 1.5B | Vanilla ICL | 4 | $66.3_{8.6}$ | $30.3_{8.9}$ | $56.5_{6.6}$ | $53.4_{8.1}$ | $34.7_{7.5}$ | $54.2_{5.5}$ | $30.8_{8.1}$ | $61.9_{8.7}$ | $54.6_{9.9}$ | $40.8_{7.8}$ |
| | IDAICL | | $87.4_{1.5}$ | $38.8_{1.7}$ | $80.9_{1.2}$ | $82.1_{2.1}$ | $48.1_{0.6}$ | $77.8_{3.0}$ | $49.5_{1.9}$ | $87.4_{2.6}$ | $79.2_{1.8}$ | $54.1_{2.7}$ |
| | Vanilla ICL | 8 | $57.2_{7.0}$ | $30.8_{6.1}$ | $64.9_{8.3}$ | $57.6_{6.4}$ | $38.6_{6.4}$ | $57.3_{10.3}$ | $39.5_{5.3}$ | $67.4_{8.1}$ | $56.3_{5.4}$ | $47.4_{5.1}$ |
| | IDAICL | | $89.5_{1.8}$ | $40.8_{1.9}$ | $82.1_{1.2}$ | $84.3_{2.1}$ | $50.2_{3.4}$ | $80.1_{2.9}$ | $51.5_{2.5}$ | $89.8_{1.7}$ | $80.3_{0.9}$ | $55.5_{0.6}$ |
| | Vanilla ICL | 12 | $70.9_{9.6}$ | $34.7_{6.7}$ | $65.2_{5.6}$ | $59.9_{6.7}$ | $38.3_{10.2}$ | $59.6_{8.1}$ | $40.7_{7.5}$ | $72.5_{11.6}$ | $57.6_{9.5}$ | $48.5_{5.7}$ |
| | IDAICL | | $90.0_{2.8}$ | $41.1_{1.3}$ | $83.4_{2.3}$ | $85.6_{2.4}$ | $51.6_{2.9}$ | $80.5_{2.5}$ | $51.8_{3.6}$ | $90.5_{2.7}$ | $81.1_{3.0}$ | $55.7_{2.1}$ |
| | MetaICL | 12 | $79.1_{7.0}$ | $38.6_{3.7}$ | $76.4_{6.3}$ | $75.3_{4.5}$ | $50.5_{7.1}$ | $73.9_{7.6}$ | $46.7_{6.3}$ | $86.8_{7.8}$ | $76.4_{5.4}$ | $53.1_{1.6}$ |
| | +IDAICL | | $89.6_{2.2}$ | $\underline{42.9}_{2.3}$ | $84.2_{3.4}$ | $\underline{87.9}_{1.1}$ | $\mathbf{53.8}_{1.2}$ | $\underline{83.4}_{3.2}$ | $53.6_{1.3}$ | $\underline{91.9}_{0.9}$ | $\underline{84.3}_{1.4}$ | $\underline{57.3}_{1.5}$ |
| | Channel ICL | 12 | $83.3_{5.9}$ | $37.5_{4.6}$ | $80.6_{4.1}$ | $77.1_{5.5}$ | $48.9_{6.7}$ | $68.2_{8.3}$ | $43.3_{7.2}$ | $70.4_{9.3}$ | $67.9_{5.5}$ | $53.6_{8.9}$ |
| | +IDAICL | | $\mathbf{91.2}_{2.1}$ | $40.8_{1.5}$ | $\underline{86.5}_{2.6}$ | $\mathbf{88.2}_{1.8}$ | $52.4_{2.9}$ | $82.3_{2.4}$ | $50.5_{1.8}$ | $88.7_{1.2}$ | $82.6_{0.9}$ | $56.5_{2.1}$ |
| | EPR | 12 | $82.8_{2.6}$ | $40.6_{2.1}$ | $79.5_{1.4}$ | $74.7_{2.7}$ | $50.7_{2.3}$ | $83.3_{0.7}$ | $\underline{82.2}_{2.4}$ | $91.5_{0.8}$ | $83.2_{1.6}$ | $54.8_{1.9}$ |
| | +IDAICL | | $\underline{90.5}_{1.5}$ | $\mathbf{43.8}_{1.0}$ | $\mathbf{87.4}_{0.9}$ | $86.5_{1.5}$ | $\underline{52.9}_{1.8}$ | $\mathbf{85.8}_{0.5}$ | $\mathbf{84.7}_{1.1}$ | $\mathbf{93.5}_{2.5}$ | $\mathbf{86.4}_{2.2}$ | $\mathbf{57.5}_{1.5}$ |
| GPT-Neo | MetaICL | 12 | $87.8_{6.7}$ | $42.5_{6.1}$ | $82.2_{5.9}$ | $80.7_{4.8}$ | $51.5_{5.3}$ | $72.2_{8.2}$ | $54.1_{6.8}$ | $84.4_{5.5}$ | $74.3_{8.2}$ | $50.3_{6.4}$ |
| | +IDAICL | | $\underline{92.1}_{1.1}$ | $44.3_{2.3}$ | $\mathbf{88.8}_{2.1}$ | $\mathbf{88.1}_{1.8}$ | $\mathbf{53.2}_{1.7}$ | $84.3_{2.1}$ | $64.3_{1.9}$ | $94.3_{1.2}$ | $86.5_{0.9}$ | $\mathbf{53.4}_{2.1}$ |
| | Channel ICL | 12 | $83.4_{5.4}$ | $39.8_{6.4}$ | $79.5_{5.7}$ | $79.4_{5.9}$ | $50.1_{3.8}$ | $70.6_{8.2}$ | $50.8_{5.1}$ | $78.3_{7.1}$ | $72.5_{6.9}$ | $48.7_{4.5}$ |
| | +IDAICL | | $91.5_{2.2}$ | $41.6_{1.8}$ | $85.4_{1.9}$ | $\underline{87.2}_{2.5}$ | $\underline{52.7}_{2.2}$ | $83.7_{1.4}$ | $62.8_{0.7}$ | $93.5_{3.3}$ | $84.6_{3.1}$ | $52.0_{1.8}$ |
| | EPR | 12 | $88.2_{1.6}$ | $\underline{45.7}_{2.2}$ | $81.8_{1.9}$ | $71.8_{2.9}$ | $49.9_{1.1}$ | $\underline{89.4}_{2.4}$ | $\underline{92.3}_{2.2}$ | $\underline{96.1}_{1.2}$ | $\underline{88.8}_{1.1}$ | $49.4_{0.7}$ |
| | +IDAICL | | $\mathbf{93.2}_{0.8}$ | $\mathbf{47.2}_{1.3}$ | $\underline{88.5}_{1.2}$ | $86.6_{2.0}$ | $52.1_{0.4}$ | $\mathbf{93.1}_{1.2}$ | $\mathbf{94.4}_{2.4}$ | $\mathbf{97.8}_{1.5}$ | $\mathbf{91.2}_{0.7}$ | $\underline{52.1}_{0.5}$ |

Table 1: Comparison results of three PLMs. Two numbers indicate the mean accuracy (%) and standard deviation over different seeds. The best and second-best results per PLM per dataset are highlighted in bold and underlined, respectively. "+IDAICL" means that the current approach is used in conjunction with IDAICL. The results for different numbers of demonstration examples (i.e., $m$ values) using the GPT-Neo model are illustrated in Figure 3.

This approach compensates for predictions of minor classes. When different answers are uniformly distributed, $\tau \log \pi_{y_j}$ exerts an equal influence on all answer types. Consequently, the final prediction is given by

$$\hat{y} = \arg\min_{y_j \in \mathcal{Y}} \tilde{P}^{\text{IDA}}_{y_j}(\tilde{\boldsymbol{x}}). \tag{16}$$

# 4 Experimental Setup

## 4.1 Models and Datasets

We evaluated the performance of IDAICL across seven large PLMs, including GPT-2 (Radford et al., 2019) (with 0.1B, 0.3B, 0.8B, and 1.5B parameters), GPT-Neo (Black et al., 2021) (with 2.7B parameters), and LLaMA (Touvron et al., 2023) (with 13B and 33B parameters). Following previous research (Min et al., 2022a; Han et al., 2023; Lu et al., 2022), our evaluation encompasses ten text classification datasets. Among these, SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), and Amazon (McAuley and Leskovec, 2013) are five sentiment classification tasks. Subj (Pang and Lee, 2004), TREC (Voorhees and Tice, 2000), DBPedia (Lehmann et al., 2015), and AGNews (Zhang et al., 2015) cater to subjectivity, question, ontology, and news classification tasks, respectively. Additionally, CB (De Marneffe et al., 2019) is utilized for natural language inference. Among these datasets, SST-5, Amazon, TREC, and CB are characterized by imbalanced training data. Details of all datasets are provided in Section A of the Appendix.

## 4.2 Compared Baselines

Besides Vanilla ICL, we compared and integrated IDAICL with three popular ICL algorithms, focusing on learning process design and demonstration retrieval. These include MetaICL (Min et al., 2022b), Channel ICL (Min et al., 2022a), and Efficient Prompt Retrieval (EPR) (Rubin et al., 2022). Moreover, we compared IDAICL with other ad-

| PLM | Method | SST-2 | SST-5 | MR | CR | Subj | TREC | DBPedia | AGNews | CB | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LLaMA 13B | Vanilla ICL | $95.6_{7.1}$ | $29.5_{6.2}$ | $90.0_{5.8}$ | $91.4_{7.4}$ | $72.9_{6.9}$ | $62.8_{9.1}$ | $80.9_{7.6}$ | $80.2_{5.9}$ | $51.5_{8.2}$ | 72.8 |
| | ConCa | $\mathbf{96.7}_{5.4}$ | $40.3_{6.2}$ | $91.7_{7.3}$ | $90.8_{4.2}$ | $79.6_{9.1}$ | $68.2_{5.6}$ | $\underline{94.3}_{4.1}$ | $\underline{85.2}_{7.5}$ | $46.6_{5.0}$ | 77.0 |
| | ProCa | $95.4_{3.8}$ | $\underline{43.4}_{5.7}$ | $90.3_{9.6}$ | $\underline{92.1}_{3.1}$ | $\underline{84.8}_{2.5}$ | $69.9_{2.1}$ | $92.5_{4.9}$ | $81.6_{3.6}$ | $51.4_{4.2}$ | $\underline{77.9}$ |
| | D-ConCa | $96.3_{3.8}$ | $42.5_{4.5}$ | $\underline{92.0}_{4.1}$ | $90.5_{2.9}$ | $82.9_{4.5}$ | $\underline{73.7}_{3.9}$ | $87.4_{7.2}$ | $82.5_{3.3}$ | $\underline{52.2}_{4.1}$ | 77.8 |
| | IDAICL | $\mathbf{96.7}_{2.5}$ | $\mathbf{47.1}_{1.1}$ | $\mathbf{93.0}_{1.9}$ | $\mathbf{93.3}_{0.8}$ | $\mathbf{87.8}_{2.3}$ | $\mathbf{76.0}_{2.6}$ | $\mathbf{94.9}_{1.0}$ | $\mathbf{87.7}_{2.4}$ | $\mathbf{59.4}_{1.9}$ | **81.8** |
| LLaMA 33B | Vanilla ICL | $95.5_{7.2}$ | $29.4_{5.6}$ | $91.7_{5.4}$ | $\underline{91.5}_{8.1}$ | $85.1_{6.0}$ | $70.9_{4.4}$ | $86.6_{4.5}$ | $76.2_{6.1}$ | $59.2_{5.3}$ | 76.2 |
| | ConCa | $\underline{95.9}_{6.5}$ | $39.1_{4.4}$ | $90.3_{7.2}$ | $91.2_{3.6}$ | $74.6_{5.7}$ | $76.7_{6.2}$ | $\underline{92.4}_{3.9}$ | $87.3_{5.7}$ | $57.9_{6.0}$ | 78.4 |
| | ProCa | $95.5_{4.2}$ | $39.2_{6.3}$ | $\underline{92.4}_{4.1}$ | $91.3_{3.5}$ | $\underline{88.3}_{2.2}$ | $64.7_{3.8}$ | $86.9_{5.1}$ | $85.8_{7.1}$ | $\underline{59.9}_{3.8}$ | 78.2 |
| | D-ConCa | $95.4_{3.8}$ | $\underline{40.7}_{4.5}$ | $92.1_{4.2}$ | $91.0_{2.9}$ | $76.4_{3.6}$ | $\mathbf{80.2}_{2.1}$ | $87.6_{4.2}$ | $\underline{87.7}_{4.3}$ | $56.5_{3.4}$ | $\underline{78.6}$ |
| | IDAICL | $\mathbf{96.5}_{1.1}$ | $\mathbf{46.8}_{2.4}$ | $\mathbf{93.6}_{1.3}$ | $\mathbf{92.3}_{3.3}$ | $\mathbf{89.3}_{2.4}$ | $\underline{79.1}_{1.5}$ | $\mathbf{95.6}_{2.3}$ | $\mathbf{88.4}_{1.9}$ | $\mathbf{64.6}_{2.8}$ | **82.9** |

Table 2: Comparison results of Macro-F1 for the LLaMA model with 13B and 33B parameters, setting $m$ to 4.

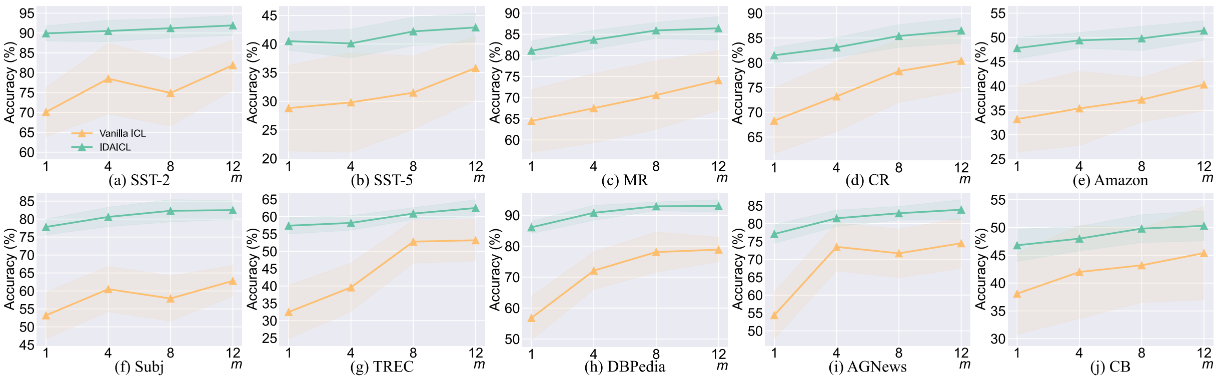



Figure 3: Comparison results between Vanilla ICL and IDAICL across different values of $m$ on the GPT-Neo model. IDAICL significantly outperforms Vanilla ICL, particularly when the number of demonstration examples is small.

vanced prediction calibration methods: Contextual Calibration (ConCa) (Zhao et al., 2021), Prototypical Calibration (ProCa) (Han et al., 2023), and Domain-Context Calibration (D-ConCa) (Fei et al., 2023). Introductions to all compared methods and comprehensive experimental settings are presented in Sections B and C of the Appendix.

# 5 Experimental Results

## 5.1 Main Results

Table 1 displays the comparison results between IDAICL and four ICL baselines (Vanilla ICL, MetaICL, Channel ICL, and EPR) across GPT-2 models (with 0.8B and 1.5B parameters) and the GPT-Neo model. These results lead to three main findings. Firstly, IDAICL consistently exhibits high effectiveness across various model sizes and datasets, highlighting its strong generalization capacity, even under scenarios involving imbalanced training data. Compared to Vanilla ICL, IDAICL outperforms by an average of 17.7% and 18.4% across diverse datasets and $m$ values for GPT-2 with 0.8B and 1.5B parameters, respectively. Secondly, in comparison to other ICL baselines like Channel ICL, MetaICL, and EPR, the integration of IDAICL consistently delivers notable performance improvements, emphasizing the efficacy of enhancing demonstrations for refined predictions. The inclusion of IDAICL led to an average performance boost of 7.3% for MetaICL and 8.2% for Channel ICL. Lastly, IDAICL notably enhances worst-case accuracy and diminishes performance variance across different seeds, showcasing its ability to improve prediction stability. Additional results on LLaMA and smaller GPT-2 models are available in Tables 7 and 8 of the Appendix.

## 5.2 Comparison with Calibration Methods

We compared IDAICL with three advanced prediction calibration methods (ConCa, ProCa, and D-ConCa) across three PLMs: GPT-2, GPT-Neo, and LLaMA. Table 2 presents the comparison results for the LLaMA models, where IDAICL consistently achieves state-of-the-art performance, except for TREC using the LLaMA model with 33B parameters. These findings suggest that IDAICL which leverages statistical information derived from the input data distribution for prediction calibration, generally outperforms methods relying on estimated biases for correction. Further comparison results can be found in Table 9 of the Appendix.

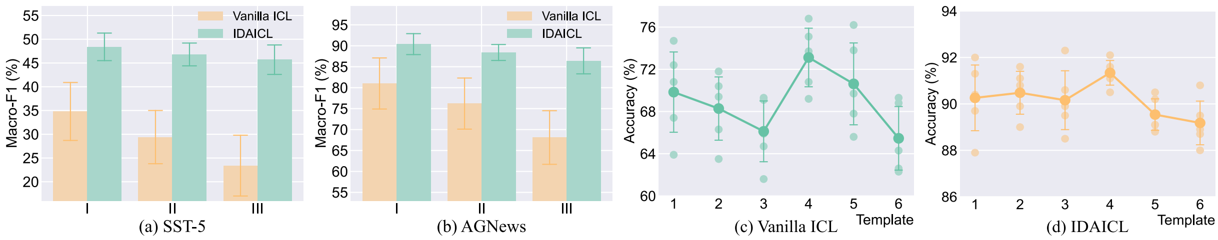



Figure 4: (a) and (b): Macro-F1 of SST-5 and AGNews datasets using the LLaMA model with 33B parameters under three demonstration selection settings, setting $m$ to 4. (c) and (d): Accuracy of Vanilla ICL and IDAICL on the SST-2 dataset using the GPT-2 model with 1.5B parameters across six templates, setting $m$ to 12. IDAICL demonstrates greater robustness across various demonstration examples and templates compared to Vanilla ICL.

### 5.3 Stability Analysis

Previous studies (Zhao et al., 2021; Sorensen et al., 2022; Min et al., 2022a; Zhang et al., 2022b) have highlighted the considerable variability in ICL's performance. In this section, we verified that IDAICL can effectively enhance performance stability across diverse scenarios.

**Varying numbers of demonstrations** We have presented the results across different numbers of demonstrations in Table 1. For a clearer depiction, the outcomes regarding GPT-Neo are illustrated in Figure 3. As the number of demonstration examples (represented by $m$) increases, both Vanilla ICL and IDAICL exhibit improved performance, emphasizing the importance of comprehensive statistical properties of the input data for IDAICL's effectiveness. Notably, IDAICL significantly enhances performance stability across various numbers of demonstrations and consistently outperforms Vanilla ICL. The performance improvement is particularly pronounced when $m$ takes on smaller values, indicating the efficacy of IDAICL in enriching the available knowledge for PLMs.

**Varying demonstrations** To confirm that augmenting demonstrations can enhance the robustness of the ICL strategy across various demonstrations, we investigated three distinct demonstration selection settings. Setting I: Training samples most similar to the test sample are chosen. Setting II: Samples are randomly selected from the training data. Setting III: Training samples exhibiting the greatest dissimilarity from the test sample are selected. As shown in Figures 4(a) and (b), IDAICL significantly outperforms Vanilla ICL and demonstrates greater robustness across the three selection settings. Additionally, our discoveries suggest that selecting demonstrations that are more similar to the test samples leads to better performance than exclusively selecting dissimilar ones, which aligns with the findings obtained by Wang et al. (2022).

**Varying templates** To assess the performance of IDAICL across various templates, we employed fifteen templates on the SST-2 dataset following those outlined by Zhao et al. (2021). The templates are elaborated in Table 10 of the Appendix. Figures 4(c) and (d) display the performance of Vanilla ICL and IDAICL across six templates. Some templates achieve higher average performance than others. Nevertheless, IDAICL consistently enhances both average and worst-case accuracy, simultaneously reducing performance variance across different templates. The complete results are available in Figure 7 of the Appendix.

**Impact of imbalance in labels** Figures 5(a) and (b) depict comparison results among Vanilla ICL, MetaICL, Channel ICL, and IDAICL across different degrees of imbalances. It is evident that the performance of Vanilla ICL is sensitive to class imbalance, while that of IDAICL and Channel ICL exhibit robustness to the imbalance. Moreover, notable performance improvements are observed with higher levels of imbalance. Additionally, Figures 5(c) and (d) illustrate the confusion matrices for CR and Subj datasets, with the proportion of one category (i.e., "Negative" and "Subjective") in demonstrations setting to 0.1 and 0.2. IDAICL significantly improves the accuracy of the underrepresented classes when compared to Vanilla ICL, thereby contributing to enhanced fairness among classes. In the subsequent section, we demonstrate that the strong performance of IDAICL in handling imbalanced label distributions stems from both the statistical properties and the class proportion term.

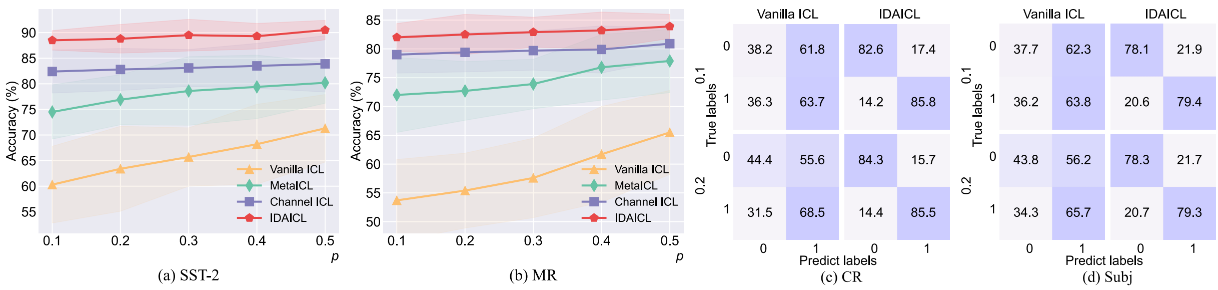



Figure 5: (a) and (b): Accuracy comparison of the SST-2 and MR datasets, where the proportions of the negative class in demonstrations (denoted as $p$) are varied from 0.1 to 0.5. (c) and (d): Confusion matrices for the CR and Subj datasets, representing scenarios where the proportions of one category in demonstrations are set to 0.1 and 0.2. The analysis is conducted using the GPT-2 model with 1.5B parameters, with $m$ setting to 12. IDAICL demonstrates greater robustness in handling imbalanced class distributions within demonstrations.

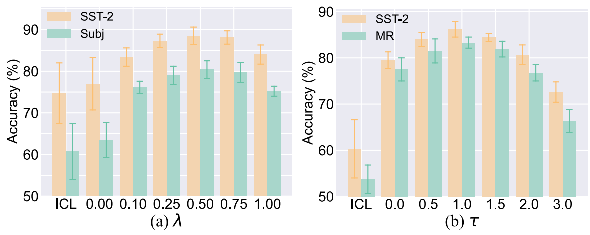



Figure 6: Accuracy across different $\lambda$ and $\tau$ values, using GPT-2 with 0.8B parameters, setting $m$ to 12. $\lambda = 0$ and $\tau = 0$ signify that the two modulating factors and the class proportion term are not utilized, respectively.

### 5.4 Sensitivity and Ablation Studies

We conducted ablation studies on IDAICL to investigate the influence of the two modulating factors and the class proportion term. The parameters $\lambda$ and $\tau$ govern the augmentation strength and the impact of the class proportion term, respectively. In Figure 6(a), a significant performance drop is observed when predictions are not calibrated using statistical properties derived from the demonstrations. Additionally, optimal performance is achieved when $\lambda$ equals 0.5.

Figure 6(b) showcases the accuracy of SST-2 and MR datasets with the negative class proportion in demonstrations setting to 0.1. Results indicate that solely leveraging statistical properties (i.e., $\tau$ equals 0) enhances performance under imbalanced demonstrations, with further improvements observed upon the inclusion of the class proportion term. Additionally, optimal performance is attained when $\tau$ equals 1. Consequently, we recommend setting $\lambda$ to 0.5 and $\tau$ to 1 for practical applications. More results are presented in Appendix F.

| Dataset | 0-shot | 1-shot | 4-shot | IDAICL |
|---|---|---|---|---|
| SST-2 | 63.2 | $61.3_{9.4}$ | $57.6_{7.1}$ | **76.3** |
| SST-5 | 25.0 | $27.3_{7.9}$ | $30.4_{6.3}$ | **33.5** |
| MR | 58.9 | $54.3_{6.8}$ | $59.3_{6.5}$ | **71.2** |
| Subj | 48.9 | $47.1_{8.3}$ | $57.6_{5.4}$ | **67.3** |

Table 3: Accuracy comparison between Vanilla ICL and IDAICL based solely on statistical properties, using the GPT-2 model with 0.8B parameters.

### 5.5 Further Discussion

To further investigate the effect of statistical properties within demonstrations on model performance, we exclusively employed queries along with statistical information for inference, excluding the inclusion of demonstrations for each test sample. These statistics were estimated using deep features of all training samples. As shown in Table 3, IDAICL relying solely on statistical properties distinctly outperforms Vanilla ICL across scenarios with zero, one, and even four demonstrations. This emphasizes the crucial role of prior statistics obtained from training data in PLMs' predictions. This phenomenon is understandable as statistical properties inherently encompass richer global information compared to individual demonstrations.

## 6 Conclusion

This study introduces IDAICL, a novel ICL approach designed to enhance demonstrations by utilizing semantic directions sampled from the deep feature distribution of demonstration examples. Our augmentation strategy enriches the knowledge available to PLMs without extending the context length. A new prediction function is then theoretically established considering the number of augmented pieces approaching infinity. This eliminates the need for explicit augmentation and allows for direct utilization of this derived function for predictions. Our extensive experiments, spanning various tasks and PLMs, demonstrate that IDAICL significantly enhances both prediction accuracy and stability when compared to other ICL baselines.

## Limitations

While IDAICL proves to be competitive in few-shot learning, there are limitations that open up avenues for future research. First, due to the necessity of accessing the parameters of the final fully connected layer in PLMs, IDAICL is exclusively suitable for open-source models. Future research is expected to develop alternative augmentation strategies tailored for black-box PLMs. Second, our evaluation of IDAICL focused on seven PLMs and ten text classification tasks. We defer further explorations involving other PLMs and non-classification tasks for future work. Additionally, IDAICL relies on a small set of demonstrations to estimate the feature mean and covariance matrix. If such a collection is unavailable or extremely scarce, IDAICL may need to be used in conjunction with demonstration generation methods.

Other avenues for future work involve exploring more effective augmentation distributions. This entails exploring finer-grained distributions, such as category-level or sample-level distributions, to emphasize the unique characteristics of individual categories or samples, and extending these distributions beyond the constraints of training data. Furthermore, given the effectiveness of data augmentation in model training, future research could explore the utilization of our derived prediction function in both the training and fine-tuning phases of large PLMs.

## References

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. 2013. Better mixing via deep representations. In *Proceedings of 30th International Conference on Machine Learning*, pages 552–560, Atlanta, Georgia, USA. ACM.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Rose Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, page 1877–1901, Online and Vancouver, Canada.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *Proceedings of the 36th Advances in Neural Information Processing Systems*, pages 18878–18891, New Orleans, USA.

Dong Chen, Yueting Zhuang, Zijin Shen, Carl Yang, Guoming Wang, Siliang Tang, and Yi Yang. 2022a. Cross-modal data augmentation for tasks of different modalities. *IEEE Transactions on Multimedia*, 99:1–11.

Hui Chen, Wei Han, Diyi Yang, and Soujanya Poria. 2022b. DoubleMix: Simple interpolation-based data augmentation for text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4622–4632, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Xiaohua Chen, Yucan Zhou, Dayan Wu, Wanqian Zhang, Yu Zhou, Bo Li, and Weiping Wang. 2022c. Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 356–364, Online. AAAI Press.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

Tsz-Him Cheung and Dit-Yan Yeung. 2021. Modals: Modality-agnostic automated data augmentation in the latent space. In *Proceedings of 9th International Conference on Learning Representations*, Online.

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835*.

Eunbi Choi, Yongrae Jo, Joel Jang, and Minjoon Seo. 2022. Prompt injection: Parameterization of fixed inputs. *arXiv preprint arXiv:2206.11349*.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23*, pages 107–124, Universitat Autònoma de Barcelona, Bellaterra.

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 (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2023. Prototypical calibration for few-shot learning of language models. In *Proceedings of 11st International Conference on Learning Representations*, Kigali, Rwanda.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*.

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*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, Washington, USA. ACM.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Junjie Li, Jianfei Yu, and Rui Xia. 2022. Generative cross-domain data augmentation for aspect and opinion co-extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4219–4229, Seattle, United States. Association for Computational Linguistics.

Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. 2023a. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*.

Shuang Li, Kaixiong Gong, Chi-Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. 2021. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5208–5217, Nashville, TN, USA. IEEE.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023b. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. 2022. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98, Hong Kong. Association for Computational Linguistics.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172, Hong Kong, China. ACM.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *Proceedings of the 9th International Conference on Learning Representations*, Vienna, Austria.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022c. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 8026–8037, Vancouver, Canada.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pages 11054–11070, Online.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

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: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia 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 (Volume 1: Long Papers)*, pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

Raphael Tang, Xinyu Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2023. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. *arXiv preprint arXiv:2310.07712*.

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*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207, Athens, Greece. ACM.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179, Dublin, Ireland. Association for Computational Linguistics.

Xindi Wang, Yufei Wang, Can Xu, Xiubo Geng, Bowen Zhang, Chongyang Tao, Frank Rudzicz, Robert E. Mercer, and Daxin Jiang. 2023. Investigating the learning behaviour of in-context learning: A comparison with supervised learning. In *Proceedings of 26th European Conference on Artificial Intelligence*, pages 2543–2551, Krakow, Poland.

Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. 2019. Implicit semantic data augmentation for deep networks. In *Proceedings of 33rd Conference on Neural Information Processing Systems*, page 12635–12644, Vancouver, Canada.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of 36th Advances in Neural Information Processing Systems*, pages 24824–24837, Online.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xing Wu, Chaochen Gao, Meng Lin, Liangjun Zang, and Songlin Hu. 2022. Text smoothing: Enhance various data augmentation methods on text classification tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 871–875, Dublin, Ireland. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1423–1436, Toronto, Canada. Association for Computational Linguistics.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, et al. 2023. Supervised knowledge makes large language models better in-context learners. In *The Twelfth International Conference on Learning Representations*.

Minjia Zhang, Niranjan Uma Naresh, and Yuxiong He. 2022a. Adversarial data augmentation for task-specific knowledge distillation of pre-trained transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11685–11693, online. AAAI Press.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, page 649–657, Montreal, Canada.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 38th International Conference on Machine Learning*, pages 12697–12706, Online. ACM.

Xiaoling Zhou and Ou Wu. 2023a. Implicit counterfactual data augmentation for deep neural networks. *arXiv preprint arXiv:2304.13431*.

Xiaoling Zhou and Ou Wu. 2023b. Which samples should be learned first: Easy or hard? *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15.

Xiaoling Zhou, Ou Wu, and Chao Jiang. 2022. Increasing naturalness of human–machine dialogue: The users' choices inference of options in machine-raised questions. *Knowledge-Based Systems*, 243:108485.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *Proceedings of the 8th International Conference on Learning Representations*, Online.

## A Details of Applied Datasets

Table 4 presents comprehensive statistics for all datasets utilized in this study. The information includes task descriptions, average sentence lengths, class counts, and details on class imbalance. Additionally, Table 5 provides sample instances and label names for each of the datasets.

## B Details of Compared Baselines

The compared methods are described as follows:

- **Vanilla ICL**: We use the PLMs as they are and implement ICL by conditioning it on a concatenation of $m$ training examples, following the approach outlined by Brown et al. (2020).
- **MetaICL**: The fundamental concept underlying MetaICL is to utilize a multi-task learning framework across a diverse range of meta-training tasks (Min et al., 2022b).
- **Channel ICL**: It employs a noisy channel approach for language model prompting in few-shot text classification (Min et al., 2022a).
- **EPR**: It employs language models to autonomously label examples that are suitable as effective prompts and subsequently trains a prompt retriever based on this acquired signal (Rubin et al., 2022).
- **ConCa**: It assesses the model's inclination towards specific answers by introducing a dummy test input that lacks content (Zhao et al., 2021).
- **ProCa**: The prediction of ProCa is calibrated based on the likelihood of prototypical clusters (Han et al., 2023).
- **D-ConCa**: It initially assesses the impacts of various label biases by employing randomly sampled words from the task corpus. During inference, it utilizes the estimated label bias to calibrate the model's output probabilities (Fei et al., 2023).

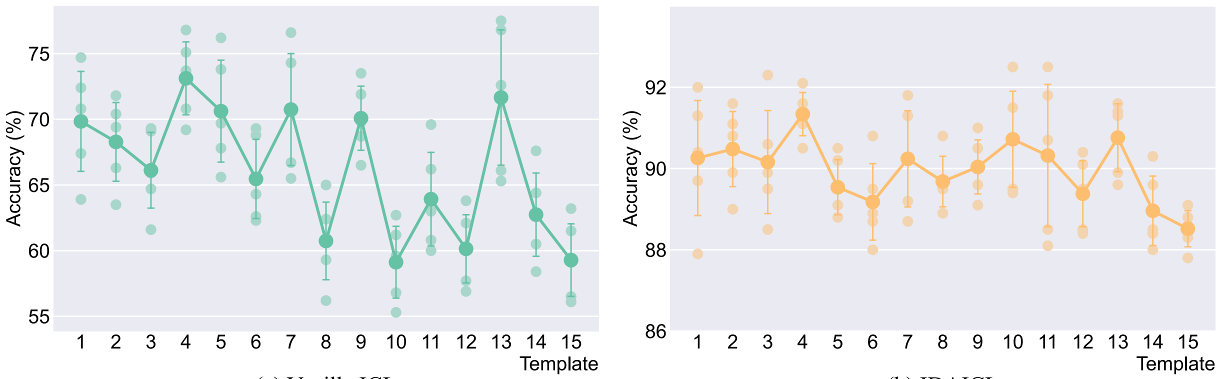
## C More Details of Experimental Settings

The entire implementation is conducted utilizing PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020). We follow the parameter configurations and details specified in previous research (Min et al., 2022a). The number of demonstrations is primarily set to $m = 12$, but we also explore $m$ values of $\{1, 4, 8, 12, 16\}$ in the ablations, with the specific settings detailed in the respective sections. Demonstration examples for each test sample are randomly selected from the training data, unless specific methods employ a specially designed selection method, such as EPR (Rubin et al., 2022). The values of the feature mean and covariance matrix are estimated from the demonstration set containing demonstration examples corresponding to all test samples. We depart from the assumption made in previous studies, which presupposes an equal distribution of training examples across all classes (Gao et al., 2021; Logan IV et al., 2022), in order to facilitate a more realistic and demanding evaluation.

Each experiment is repeated under five different random seeds. The batch size is set to 32, and the sequence length is configured to 128 for datasets with shorter texts, including SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), and TREC (Voorhees and Tice, 2000). On the other hand, for datasets with longer input texts, including AGNews (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), Subj (Pang and Lee, 2004), CB (De Marneffe et al., 2019), and Amazon (McAuley and Leskovec, 2013), a batch size of 16 and a sequence length of 256 are employed. Regarding the hyperparameters in IDAICL, the values of $\lambda$ and $\tau$ are fixed at 0.5 and 1, respectively, except in sensitivity tests. The settings used for the compared methods adhere to those specified in the original papers (Min et al., 2022a,b; Rubin et al., 2022; Zhao et al., 2021; Han et al., 2023; Fei et al., 2023). Accuracy serves as the primary evaluation metric, alongside the provided values of Macro-F1 for the LLaMA model. For each task, a specific template is utilized for inference, as detailed in Table 6. Additionally, we also examine the impact of different templates on the performance of IDAICL following those outlined by Zhao et al. (2021), which include question-answer templates, conversation-style templates, prompts resembling web pages, and variations on label names,

| Dataset | Task | Avg. length | Classes | Balanced |
|---|---|---|---|---|
| SST-2 (Socher et al., 2013) | Sentiment analysis | 12.4 | 2 | Yes |
| SST-5 (Socher et al., 2013) | Sentiment analysis | 23.1 | 5 | No |
| MR (Pang and Lee, 2005) | Sentiment analysis | 25.7 | 2 | Yes |
| CR (Hu and Liu, 2004) | Sentiment analysis | 22.1 | 2 | Yes |
| Amazon (McAuley and Leskovec, 2013) | Sentiment analysis | 78.5 | 5 | No |
| Subj (Pang and Lee, 2004) | Subjectivity classification | 28.9 | 2 | Yes |
| TREC (Voorhees and Tice, 2000) | Question classification | 11.6 | 6 | No |
| DBPedia (Lehmann et al., 2015) | Ontology classification | 65.5 | 14 | Yes |
| AGNews (Zhang et al., 2015) | News classification | 53.8 | 4 | Yes |
| CB (De Marneffe et al., 2019) | Natural language inference | 69.7/8.4 | 3 | No |

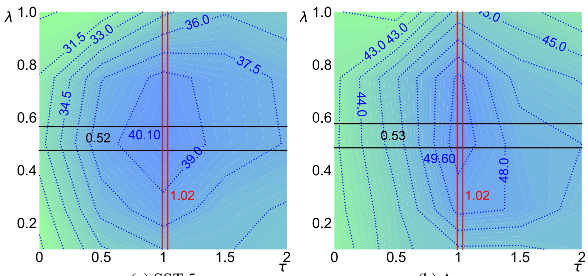
Table 4: Statistical information of ten datasets. The average length is calculated based on the GPT-2 sentence-piece length. For tasks involving sentence pairs, we provide the average length for each individual sentence.




(a) Vanilla ICL

(b) IDAICL

Figure 7: Comparison results between Vanilla ICL and IDAICL across fifteen templates. The evaluation is conducted using the GPT-2 model with 1.5B parameters. The performance of IDAICL exceeds that of Vanilla ICL and demonstrates greater robustness across various templates.




(a) SST-5

(b) Amazon

Figure 8: Results of sensitivity tests for two hyperparameters within IDAICL, i.e., $\lambda$ and $\tau$, using the GPT-2 model with 0.8B parameters, with $m$ setting to 12. Optimal performance is achieved when $\lambda \approx 0.5$ and $\tau \approx 1$.

as listed in Table 10.

## D More Comparison Results

The comparison results between Vanilla ICL and IDAICL on LLaMA models with 13B and 33B parameters across various datasets are presented in Table 7. Additionally, the corresponding results for GPT-2 models with 0.1B and 0.3B parameters are outlined in Table 8. It is evident that IDAICL consistently outperforms Vanilla ICL across all datasets and different model sizes, highlighting the high generalization capability of IDAICL. Additionally, IDAICL showcases reduced performance variance and significantly enhances the worst-case performance. Based on the findings presented in Table 9, IDAICL generally outperforms other prediction calibration methods, demonstrating the significance of statistical properties derived from the input data distribution in the predictions of PLMs.

## E More Results for Varying Templates

The comparison results between Vanilla ICL and IDAICL under all fifteen prompt templates are presented in Figure 7, illustrating that IDAICL consistently enhances both average and worst-case accuracy across all templates. Furthermore, the performance variance of IDAICL among different templates is notably smaller when compared to Vanilla ICL, highlighting the robustness of IDAICL's performance across diverse templates.

| Dataset | Instances | Label names |
|---|---|---|
| SST-2 | 1. This movie is amazing! (Label = "Positive")<br>2. Horrific movie, don't see it. (Label = "Negative") | Positive, Negative |
| SST-5 | 1. A pretensions – and disposable story — sink the movie. (Label = "Great")<br>2. Apparently reassembled from the cutting-room floor of any given daytime soap. (Label = "Terrible") | Terrible, Bad, Okay, Good, Great |
| MR | 1. Lame sweet home leaves no southern stereotype unturned. (Label = "Negative")<br>2. Not so much farcical as sour. (Label = "Negative") | Negative, Positive |
| CR | 1. It takes excellent pics and is very easy to use, if you read the manual. (Label = "Negative")<br>2. Bluetooth does not work on this phone. (Label = "Negative") | Negative, Positive |
| Amazon | 1. Don't waste your money if you already have 2003... There isn't one reason to get this update if you already have MS Money 2003 Deluxe and Business. (Label ="Terrible")<br>2. The game was in perfect condition! came before it said it should have by 2 days!! I love the game and I suggest it to a lot of my friends! (Label ="Great") | Terrible, Bad, Okay, Good, Great |
| Subj | 1. This is a story about the warm relationship between a little girl and her father despite the difficult conditions they have to live in. (Label = "Objective")<br>2. Too slow, too boring, and occasionally annoying. (Label = "Subjective") | Subjective, Objective |
| TREC | 1. When did the neanderthal man live? (Label = "Number")<br>2. How do you get a broken cork out of a bottle? (Label = "Description") | Description, Entity, Expression, Human, Location, Number |
| DBPedia | 1. CMC Aviation is a charter airline based in Nairobi Kenya. (Label = "Company")<br>2. Dialectica aemula is a moth of the Gracillariidae family. (Label = "Animal") | Company, School, Artist, Athlete, Politics, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book |
| AGNews | 1. Walk in park for Yankees Drained by a difficult week, the New York Yankees needed an uplifting victory. (Label = "Sports")<br>2. NASA Mountain View claims world's fastest computer. (Label = "Technology") | World, Sports, Business, Technology |
| CB | 1. It was a complex language. Not written down but handed down. One might say it was peeled down.<br>The language was peeled down.<br>(Label = "True")<br>2. "Do you mind if I use your phone?" Ronni could see that Guido's brain was whirring.<br>Guido's brain was whirring.<br>(Label = "True") | True, False, Neither |

Table 5: Examples and label names from all datasets.

## F More Sensitivity and Ablation Studies

We performed sensitivity tests on two hyperparameters within IDAICL: $\lambda$ and $\tau$. These values govern the strength of implicit augmentation and the influence of the class proportion term, respectively. As depicted in Figure 8, optimal performance is achieved when $\lambda \approx 0.5$ and $\tau \approx 1$ for both datasets. Furthermore, Figures 9(a) and (b) illustrate the average performance of ten datasets across different hyperparameter settings. Much like the earlier findings, the best average performance is identified at $\lambda = 0.5$ and $\tau = 1$. Consequently, setting $\lambda$ as 0.5 and $\tau$ as 1 is recommended for real applications. Furthermore, the performance remains stable within the ranges of $\lambda \in \{0.25, 0.5, 0.75\}$ and $\tau \in \{0.5, 1, 1.5\}$, indicating that adjustments can be made within these stable ranges.

## G More Results for Imbalanced Labels

The imbalanced label distribution in the training data has a significant impact on the classification performance of the model (Zhou and Wu, 2023b; Zhou et al., 2022). We depicted the confusion matrices for the SST-2 and MR datasets under two imbalance levels in Figures 9(c) and (d), in which the proportion of the negative class in demonstrations is set to 0.1 and 0.2. These results manifest that IDAICL significantly enhances the performance of the underrepresented classes in comparison to

| Dataset | Template | Label mapping |
|---|---|---|
| SST-2 | Review: {Sentence}<br>Sentiment: {Label} | Positive / Negative |
| SST-5 | Review: {Sentence}<br>Sentiment: {Label} | terrible / bad / okay / good / great |
| MR | Review: {Sentence}<br>Sentiment: {Label} | Positive / Negative |
| CR | Review: {Sentence}<br>Sentiment: {Label} | Positive / Negative |
| Subj | Input: {Sentence}<br>Type: {Label} | objective / subjective |
| TREC | Question: {Sentence}<br>Type: {Label} | description / entity / expression / human / location / number |
| Amazon | Review: {Sentence}<br>Sentiment: {Label} | terrible / bad / okay / good / great |
| AGNews | Input: {Sentence}<br>Type: {Label} | world / sports / business / technology |
| DBPedia | Input: {Sentence}<br>Type: {Label} | company / school / artist / athlete / politics / transportation<br>building / nature / village / animal / plant / album / film / book |
| CB | Premise: {Sentence}<br>Hypothesis: {Sentence}<br>Prediction: {Label} | true / false / neither |

Table 6: Prompt templates and label mappings for each dataset.

| PLM | Method | m | SST-2 | SST-5 | MR | CR | Subj | TREC | DBPedia | AGNews | CB | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13B | Vanilla ICL | 4 | $95.6_{7.1}$ | $29.5_{6.2}$ | $90.0_{5.8}$ | $91.4_{7.4}$ | $72.9_{6.9}$ | $62.8_{9.1}$ | $80.9_{7.6}$ | $80.2_{5.9}$ | $51.5_{8.2}$ | 72.8 |
| | IDAICL | | $96.7_{2.5}$ | $47.1_{1.1}$ | $93.0_{1.9}$ | $93.3_{0.8}$ | $87.8_{2.3}$ | $76.0_{2.6}$ | $94.9_{1.0}$ | $87.7_{2.4}$ | $59.4_{1.9}$ | 81.8 |
| | Vanilla ICL | 8 | $96.7_{7.1}$ | $39.4_{5.6}$ | $92.3_{6.2}$ | $92.2_{4.8}$ | $70.8_{5.1}$ | $71.2_{9.1}$ | $83.7_{4.2}$ | $79.5_{6.3}$ | $52.4_{3.7}$ | 75.4 |
| | IDAICL | | $96.9_{2.1}$ | $49.2_{1.9}$ | $93.4_{1.6}$ | $92.9_{1.9}$ | $87.5_{3.0}$ | $79.9_{2.1}$ | $93.6_{0.9}$ | $88.0_{1.7}$ | $62.4_{2.5}$ | 82.6 |
| 33B | Vanilla ICL | 4 | $95.5_{7.2}$ | $29.4_{5.6}$ | $91.7_{5.4}$ | $91.5_{8.1}$ | $85.1_{6.0}$ | $70.9_{4.4}$ | $86.6_{4.5}$ | $76.2_{6.1}$ | $59.2_{5.3}$ | 76.2 |
| | IDAICL | | $96.5_{1.1}$ | $46.8_{2.4}$ | $93.6_{1.3}$ | $92.3_{3.3}$ | $89.3_{2.4}$ | $79.1_{1.5}$ | $95.6_{2.3}$ | $88.4_{1.9}$ | $64.6_{2.8}$ | 82.9 |
| | Vanilla ICL | 8 | $96.8_{7.3}$ | $34.3_{5.4}$ | $93.4_{5.8}$ | $92.7_{6.4}$ | $83.5_{5.5}$ | $66.9_{4.8}$ | $84.1_{6.2}$ | $84.7_{5.5}$ | $62.0_{5.2}$ | 77.6 |
| | IDAICL | | $96.9_{2.3}$ | $50.3_{1.5}$ | $93.9_{2.2}$ | $93.0_{1.4}$ | $89.0_{1.0}$ | $83.1_{1.7}$ | $95.9_{2.0}$ | $88.0_{1.2}$ | $70.4_{1.8}$ | 84.5 |

Table 7: Comparison results of Macro-F1 between Vanilla ICL and IDAICL under varying values of $m$ on the LLaMA models with 13B and 33B parameters.

Vanilla ICL, thus proving its capability to address the class imbalance in demonstrations.

## H Varying Demonstration Permutations

Research has substantiated that the performance of ICL is sensitive to the permutation of demonstrations (Lu et al., 2022; Zhao et al., 2021). We assessed the performance of IDAICL under varying demonstration permutations. Specifically, we selected ten different sets of twelve training examples from the SST-2 datasets. For each set of examples, we shuffled the order ten times and calculated the accuracy for each permutation. The findings are depicted in Figure 10, indicating that IDAICL exhibits relatively stable performance across different demonstrations and permutations, while Vanilla ICL demonstrates high variance.

| PLM | Method | m | SST-2 | SST-5 | MR | CR | Amazon | Subj | TREC | DBPedia | AGNews | CB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPT-2 0.1B | Vanilla ICL | 4 | 56.3$_{7.1}$ | 28.4$_{8.8}$ | 55.4$_{7.4}$ | 54.2$_{6.2}$ | 30.8$_{8.4}$ | 52.9$_{7.9}$ | 32.2$_{5.1}$ | 44.3$_{6.2}$ | 42.8$_{9.3}$ | 42.1$_{9.6}$ |
| | IDAICL | | 69.5$_{2.6}$ | 35.3$_{1.1}$ | 66.4$_{2.3}$ | 67.2$_{2.7}$ | 39.3$_{2.9}$ | 57.2$_{2.6}$ | 44.3$_{1.8}$ | 62.2$_{2.3}$ | 65.5$_{2.7}$ | 49.2$_{1.9}$ |
| | Vanilla ICL | 8 | 60.8$_{8.3}$ | 30.6$_{6.9}$ | 57.5$_{9.7}$ | 56.0$_{5.1}$ | 33.6$_{7.8}$ | 53.7$_{5.6}$ | 33.0$_{10.7}$ | 52.1$_{5.8}$ | 45.6$_{9.1}$ | 45.4$_{6.2}$ |
| | IDAICL | | 71.4$_{1.8}$ | 36.1$_{2.9}$ | 67.6$_{1.8}$ | 68.6$_{2.2}$ | 40.0$_{0.7}$ | 58.5$_{2.5}$ | 45.6$_{1.9}$ | 63.6$_{1.1}$ | 66.9$_{1.6}$ | 50.6$_{2.7}$ |
| | Vanilla ICL | 12 | 64.5$_{6.0}$ | 30.8$_{7.1}$ | 59.3$_{5.6}$ | 59.1$_{8.4}$ | 33.9$_{5.5}$ | 56.6$_{8.9}$ | 35.8$_{7.1}$ | 52.3$_{11.4}$ | 47.4$_{6.0}$ | 47.4$_{7.7}$ |
| | IDAICL | | 72.2$_{1.1}$ | 36.7$_{2.2}$ | 70.1$_{1.7}$ | 69.3$_{1.8}$ | 40.8$_{1.2}$ | 60.9$_{1.5}$ | 47.0$_{2.7}$ | 65.5$_{1.9}$ | 67.8$_{2.2}$ | 51.2$_{3.3}$ |
| | Vanilla ICL | 16 | 64.3$_{6.1}$ | 33.5$_{7.1}$ | 59.9$_{6.6}$ | 61.7$_{7.5}$ | 34.6$_{6.9}$ | 56.1$_{6.2}$ | 36.9$_{5.7}$ | 54.1$_{7.2}$ | 47.9$_{8.0}$ | 48.9$_{7.7}$ |
| | IDAICL | | 72.9$_{2.5}$ | 38.0$_{2.4}$ | 69.7$_{1.3}$ | 69.9$_{2.1}$ | 41.7$_{0.9}$ | 60.6$_{1.1}$ | 46.6$_{1.9}$ | 65.9$_{2.6}$ | 65.7$_{1.0}$ | 51.8$_{2.2}$ |
| GPT-2 0.3B | Vanilla ICL | 4 | 60.8$_{7.5}$ | 26.6$_{6.8}$ | 50.5$_{7.1}$ | 52.3$_{6.1}$ | 30.5$_{5.2}$ | 53.2$_{8.3}$ | 32.8$_{8.1}$ | 50.5$_{4.8}$ | 41.3$_{5.9}$ | 42.7$_{7.1}$ |
| | IDAICL | | 78.4$_{1.7}$ | 33.1$_{2.5}$ | 66.6$_{0.9}$ | 70.3$_{2.3}$ | 40.1$_{1.5}$ | 69.4$_{1.7}$ | 45.6$_{3.3}$ | 66.2$_{2.1}$ | 62.8$_{3.7}$ | 50.4$_{1.8}$ |
| | Vanilla ICL | 8 | 58.9$_{8.7}$ | 29.4$_{6.1}$ | 52.4$_{8.9}$ | 54.8$_{8.2}$ | 32.7$_{7.9}$ | 53.5$_{6.7}$ | 34.0$_{8.2}$ | 59.1$_{9.7}$ | 43.8$_{6.4}$ | 46.9$_{7.6}$ |
| | IDAICL | | 80.8$_{1.7}$ | 34.8$_{1.9}$ | 69.5$_{1.1}$ | 71.5$_{0.8}$ | 41.5$_{1.7}$ | 70.3$_{2.6}$ | 46.2$_{2.2}$ | 68.1$_{1.7}$ | 63.3$_{2.1}$ | 51.5$_{2.5}$ |
| | Vanilla ICL | 12 | 62.9$_{14.4}$ | 30.6$_{7.8}$ | 55.2$_{6.2}$ | 56.1$_{6.7}$ | 34.2$_{7.5}$ | 56.8$_{7.1}$ | 36.2$_{9.8}$ | 58.0$_{7.3}$ | 46.5$_{9.3}$ | 48.6$_{6.6}$ |
| | IDAICL | | 82.2$_{2.3}$ | 36.1$_{1.8}$ | 68.9$_{2.4}$ | 72.0$_{1.5}$ | 43.7$_{0.6}$ | 71.4$_{2.4}$ | 48.3$_{1.3}$ | 70.5$_{1.9}$ | 65.2$_{2.2}$ | 52.9$_{1.4}$ |
| | Vanilla ICL | 16 | 67.4$_{6.3}$ | 31.7$_{7.1}$ | 57.6$_{8.6}$ | 56.6$_{5.2}$ | 34.7$_{6.2}$ | 57.0$_{5.3}$ | 38.1$_{6.9}$ | 59.3$_{8.2}$ | 45.2$_{7.6}$ | 49.4$_{8.7}$ |
| | IDAICL | | 81.5$_{2.8}$ | 36.8$_{1.2}$ | 70.4$_{1.7}$ | 72.9$_{2.1}$ | 43.1$_{1.3}$ | 71.9$_{2.7}$ | 48.7$_{1.1}$ | 70.9$_{2.9}$ | 65.8$_{1.2}$ | 52.4$_{1.8}$ |

Table 8: Accuracy comparison between Vanilla ICL and IDAICL under varying values of $m$ on the GPT-2 models with 0.1B and 0.3B parameters.

| PLM | Method | SST-5 | MR | AGNews | TREC | SST-2 | Subj | DBPedia | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| GPT-2 1.5B | Vanilla ICL | 30.8$_{6.1}$ | 64.9$_{8.3}$ | 57.5$_{6.7}$ | 40.4$_{5.1}$ | 57.2$_{7.0}$ | 57.3$_{10.3}$ | 67.6$_{7.5}$ | 53.7 |
| | ConCa | 32.8$_{7.1}$ | 74.5$_{5.1}$ | 62.7$_{6.1}$ | 45.8$_{2.5}$ | 73.9$_{8.6}$ | 68.3$_{7.4}$ | 75.0$_{4.0}$ | 61.9 |
| | ProCa* | $\underline{36.5}_{4.4}$ | 80.8$_{6.4}$ | 75.5$_{3.2}$ | 46.0$_{2.5}$ | $\underline{88.0}_{1.3}$ | **80.2**$_{3.3}$ | $\underline{89.4}_{0.7}$ | $\underline{70.9}$ |
| | D-ConCa | 31.7$_{3.3}$ | $\underline{80.9}_{3.7}$ | $\underline{77.0}_{4.1}$ | $\underline{47.1}_{2.8}$ | 86.5$_{4.4}$ | 76.8$_{5.2}$ | 86.1$_{6.3}$ | 69.4 |
| | IDAICL | **40.8**$_{1.9}$ | **82.1**$_{1.2}$ | **80.8**$_{2.4}$ | **52.0**$_{2.5}$ | **89.5**$_{1.8}$ | $\underline{80.1}_{2.9}$ | **91.0**$_{2.5}$ | **73.8** |
| GPT-Neo | Vanilla ICL | 31.5$_{6.4}$ | 70.6$_{8.1}$ | 71.9$_{6.8}$ | 53.0$_{6.9}$ | 74.9$_{8.3}$ | 57.9$_{6.3}$ | 78.5$_{6.5}$ | 62.6 |
| | ConCa | 33.9$_{4.3}$ | 78.2$_{5.3}$ | 73.6$_{3.8}$ | 55.9$_{7.2}$ | 82.0$_{9.5}$ | 71.3$_{6.4}$ | 90.0$_{3.6}$ | 69.3 |
| | ProCa* | $\underline{39.4}_{4.0}$ | 77.8$_{13.9}$ | 78.9$_{2.5}$ | 56.0$_{3.6}$ | **91.9**$_{1.2}$ | $\underline{81.3}_{3.8}$ | $\underline{92.0}_{1.5}$ | $\underline{73.9}$ |
| | D-ConCa | 32.9$_{4.1}$ | $\underline{84.6}_{2.8}$ | $\underline{81.2}_{3.9}$ | $\underline{57.6}_{4.7}$ | $\underline{91.6}_{5.3}$ | 70.9$_{2.9}$ | 85.7$_{3.1}$ | 72.1 |
| | IDAICL | **42.2**$_{2.5}$ | **85.9**$_{1.6}$ | **83.1**$_{1.9}$ | **61.4**$_{1.7}$ | 91.2$_{2.4}$ | **82.3**$_{3.1}$ | **93.0**$_{1.5}$ | **77.0** |

Table 9: Accuracy comparison between IDAICL and other prediction calibration approaches using the GPT-2 (with 1.5B parameters) and GPT-Neo models, with $m$ setting to 8. The templates used align with those utilized by Han et al. (2023). $*$ indicates that the results were derived from the original paper.

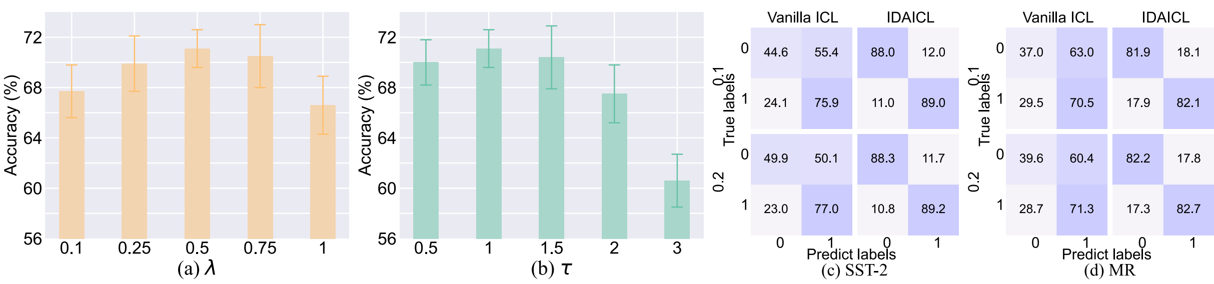
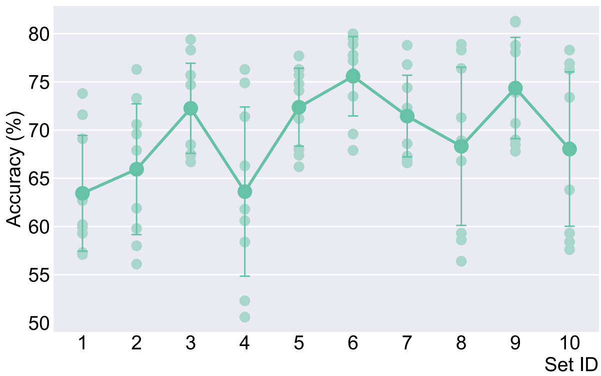
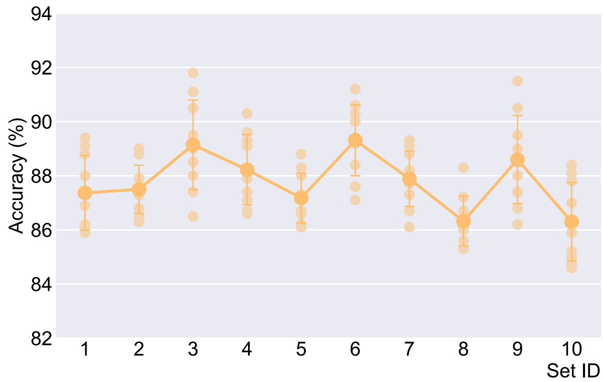



Figure 9: (a) and (b): Average accuracy across ten datasets for various values of $\lambda$ and $\tau$. Optimal average performance is attained when $\lambda = 0.5$ and $\tau = 1$. (c) and (d): Confusion matrices for the SST-2 and MR datasets under two levels of imbalance, where the proportions of the negative class in demonstrations are set to 0.1 and 0.2, respectively. When compared to Vanilla ICL, IDAICL improves the performance of the minor class. These experiments are conducted on the GPT-2 model with 1.5B parameters, setting $m$ to 12.




(a) Vanilla ICL




(b) IDAICL

Figure 10: Comparison results between Vanilla ICL and IDAICL across various demonstrations and permutations. The GPT-2 model with 0.8B parameters is employed for this analysis, setting $m$ to 12. IDAICL exhibits smaller performance variance across different demonstrations and permutations compared to Vanilla ICL.

| Format ID | Prompt | Label names |
|---|---|---|
| 1 | Review: This movie is amazing!<br>Answer: Positive<br>Review: Horrific movie, don't see it.<br>Answer: | Positive / Negative |
| 2 | Review: This movie is amazing!<br>Answer: good<br>Review: Horrific movie, don't see it.<br>Answer: | good / bad |
| 3 | My review for last night's film: This movie is amazing! The critics agreed that this movie was good<br>My review for last night's film: Horrific movie, don't see it. The critics agreed that this movie was | good / bad |
| 4 | Here is what our critics think for this month's films.<br>One of our critics wrote "This movie is amazing!". Her sentiment towards the film was positive.<br>One of our critics wrote "Horrific movie, don't see it". Her sentiment towards the film was | positive / negative |
| 5 | Critical reception [ edit ]<br>In a contemporary review, Roger Ebert wrote "This movie is amazing!". Entertainment Weekly agreed, and the overall critical reception of the film was good.<br>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!<br>Positive Review? Yes<br>Review: Horrific movie, don't see it.<br>Positive Review? | Yes / No |
| 7 | Review: This movie is amazing!<br>Question: Is the sentiment of the above review Positive or Negative?<br>Answer: Positive<br>Review: Horrific movie, don't see it.<br>Question: Is the sentiment of the above review Positive or Negative?<br>Answer: | Positive / Negative |
| 8 | Review: This movie is amazing!<br>Question: Did the author think that the movie was good or bad?<br>Answer: good<br>Review: Horrific movie, don't see it.<br>Question: Did the author think that the movie was good or bad?<br>Answer: | good / bad |
| 9 | Question: Did the author of the following tweet think that the movie was good or bad?<br>Tweet: This movie is amazing!<br>Answer: good<br>Question: Did the author of the following tweet think that the movie was good or bad?<br>Tweet: Horrific movie, don't see it<br>Answer: | good / bad |
| 10 | This movie is amazing! My overall feeling was that the movie was good<br>Horrific movie, don't see it. My overall feeling was that the movie was | good / bad |
| 11 | This movie is amazing! I liked the movie.<br>Horrific movie, don't see it. I | liked / hated |
| 12 | This movie is amazing! My friend asked me if I would give the movie 0 or 5 stars, I said 5<br>Horrific movie, don't see it. My friend asked me if I would give the movie 0 or 5 stars, I said | 0 / 5 |
| 13 | Input: This movie is amazing!<br>Sentiment: Positive<br>Input: Horrific movie, don't see it.<br>Sentiment: | Positive / Negative |
| 14 | Review: This movie is amazing!<br>Positive: True<br>Review: Horrific movie, don't see it.<br>Positive: | True / False |
| 15 | Review: This movie is amazing!<br>Stars: 5<br>Review: Horrific movie, don't see it.<br>Stars: | 5 / 0 |

Table 10: The templates employed for examining the influence of formats on the SST-2 dataset, following those outlined by Zhao et al. (2021). An example from the training data is used for illustration.