RD-MCSA: A Multi-Class Sentiment Analysis Approach Integrating In-Context Classification Rationales and Demonstrations

Anonymous ACL submission

Abstract

This paper addresses the important yet underexplored task of **multi-class sentiment analysis (MCSA)**, which remains challenging due to subtle semantic differences between adjacent sentiment categories and the scarcity of high-quality annotated data. To tackle these challenges, **RD-MCSA** (**R**ationales and **D**emonstrations-based **M**ulti-**C**lass **S**entiment **A**nalysis) is proposed as an In-Context Learning (ICL) framework designed to improve MCSA performance under limited supervision by integrating classification rationales and adaptively selected demonstrations. First, semantically grounded classification rationales are generated from a representative, class-balanced subset of annotated samples selected using a tailored balanced coreset algorithm. These rationales are then paired with demonstrations selected via a similarity-based mechanism powered by a **multi-kernel Gaussian process (MK-GP)**, enabling large language models (LLMs) to better capture fine-grained sentiment distinctions. Experiments on five benchmark sentiment datasets show that RD-MCSA consistently outperforms both supervised baselines and standard ICL methods across various evaluation metrics.

1 Introduction

Multi-Class Sentiment Analysis (MCSA) extends beyond basic sentiment polarity classification (e.g., positive or negative) by distinguishing varying levels of emotional intensity (e.g., differentiating between “very positive” and “generally positive”). By capturing finer sentiment distinctions, MCSA enables deeper insights into sentiment expression, making it essential for applications requiring fine-grained sentiment analysis (Wang et al., 2023). For instance, in opinion dynamics research, an essential step involves categorizing users’ natural language expressions into five or more sentiment or opinion categories (Chuang et al., 2024).

Despite its importance, MCSA remains challenging due to subtle semantic differences between adjacent sentiment levels, which are often difficult to distinguish accurately (Mamta and Ekbal, 2023). Additionally, sentiment categorization criteria can vary considerably across domains and applications (Rosenthal et al., 2019), further complicating the modeling process. Addressing a new MCSA task typically requires a substantial amount of high-quality, task-specific annotated data (Krosuri and Aravapalli, 2023), which are frequently limited in low-resource settings.

Large Language Models (LLMs) have demonstrated strong performance in sentiment analysis, making them a promising tool for MCSA. However, while LLMs perform well in basic sentiment classification, they often struggle with nuanced distinctions between adjacent sentiment categories (Zhang et al., 2024). In-Context Learning (ICL), which enhances LLM capabilities through task demonstrations, has achieved state-of-the-art performance across various NLP tasks. Nevertheless, its application to classification scenarios involving multiple sentiment categories remains underexplored (Randl et al., 2024). Our experimental results further indicate that conventional ICL approaches are insufficient for effectively handling the complexity of MCSA.

To address these limitations, this paper proposes **RD-MCSA**, a novel framework aimed at improving ICL performance for MCSA. RD-MCSA refines the two core components of ICL—*prompt design* and *demonstration selection*—by incorporating **classification rationales** and an **adaptive example selection** mechanism. This design enables LLMs to better capture fine-grained sentiment distinctions and improve classification accuracy.

The main contributions of this paper are summarized as follows:

1. **Rationale-Augmented ICL**: An ICL frame-

work that integrates classification rationales and demonstration examples is proposed, enabling LLMs to more effectively capture fine-grained sentiment distinctions in MCSA.

2. **Classification Rationale Generation via Tailored Balanced Coreset**: A rationale generation strategy is designed that guides LLMs to produce linguistically and semantically rich classification rationales, based on representative and class-balanced samples selected through a tailored balanced Coreset algorithm.

3. **Adaptive Demonstration Selection via MK-GP**: A novel demonstration selection method based on a multi-kernel Gaussian process (MK-GP) is proposed, enabling adaptive similarity modeling beyond fixed metrics such as cosine similarity, marking the first use of kernel-based selection in the ICL setting.

A series of comprehensive experiments conducted on five diverse and representative datasets validate the effectiveness of RD-MCSA, highlighting its advantages and identifying key challenges in MCSA tasks.

2 Related Work

2.1 Multi-class Sentiment Analysis

Multi-class sentiment analysis (MCSA), also referred to as fine-grained or graded sentiment analysis (Sharma et al., 2024), extends traditional sentiment classification by categorizing sentiments into multiple distinct classes. It refines sentiment intensity beyond basic polarity classification (e.g., "positive"/"negative") by introducing subcategories such as "very positive" and "slightly positive," or by adopting rating scales (e.g., 1–5) (AlQahtani, 2021). This provides a more nuanced understanding of sentiment in text.

Traditional MCSA models rely on supervised machine learning (Wang et al., 2023) and are commonly applied to texts such as tweets, movie reviews, and product reviews. In many cases, sentiment analysis focuses on specific targets or aspects. Widely used MCSA datasets include SemEval-2017 Task 4 (Rosenthal et al., 2019), SST-5 (Socher et al., 2013), and Amazon Reviews (AlQahtani, 2021).

Another research direction treats sentiment intensity assessment as a regression problem, where sentiment is predicted on a continuous scale. Notable tasks and datasets include SemEval-2017 Task 5 (Cortis et al., 2017), FiQA 2018 (de França Costa and da Silva, 2018), and recent dimABSA tasks at SIGHAN-2024 (Lee et al., 2024).

Despite ongoing advances, MCSA still faces key challenges, such as limited classification accuracy and the high cost of large-scale annotation—especially as sentiment granularity increases (Krosuri and Aravapalli, 2023). Fine-grained sentiment analysis for specific entities often requires distinct annotated datasets, making large-scale deployment impractical.

To address these challenges, this study aims to enhance MCSA performance under limited labeled data conditions, while maintaining broad applicability across diverse MCSA scenarios.

2.2 Text Analysis Using LLMs

Large-scale language models outperform smaller models across many NLP tasks, especially when annotation resources are limited (Zhang et al., 2024), making them a promising solution for MCSA.

Recent research on LLM-based text analysis has focused on in-context learning, where carefully selected demonstration examples guide the model's predictions. Common strategies for selecting examples include similarity-based selection (Liu et al., 2022), diversity-based selection (Levy et al., 2023), LLM feedback (Shi et al., 2022), information-theoretic criteria (Wu et al., 2023), task-level selection (Li and Qiu, 2023), active learning (Zhang et al., 2022a), and contrastive learning (Chen et al., 2024). For MCSA, a recent study (Chuang et al., 2024) applies similarity-based demonstration selection within ICL to analyze opinion dynamics.

Despite their potential, LLMs still face challenges in many NLP tasks. While effective for simpler tasks, they struggle with nuanced sentiment analysis (Zhang et al., 2024). Additionally, few-shot ICL requires further research on optimal prompt design (Liu et al., 2022). To our knowledge, no prior work has explored few-shot prompting for multi-class prediction with a large number of classes (Randl et al., 2024). Long prompts may overload LLMs (Liu et al., 2024), and context window limitations may restrict the effective representation of all classes.

This study focuses on two key components of ICL—prompt construction and demonstration selection, addressing how to **effectively provide classification information to LLMs** and how to adapt both components to better serve MCSA tasks.

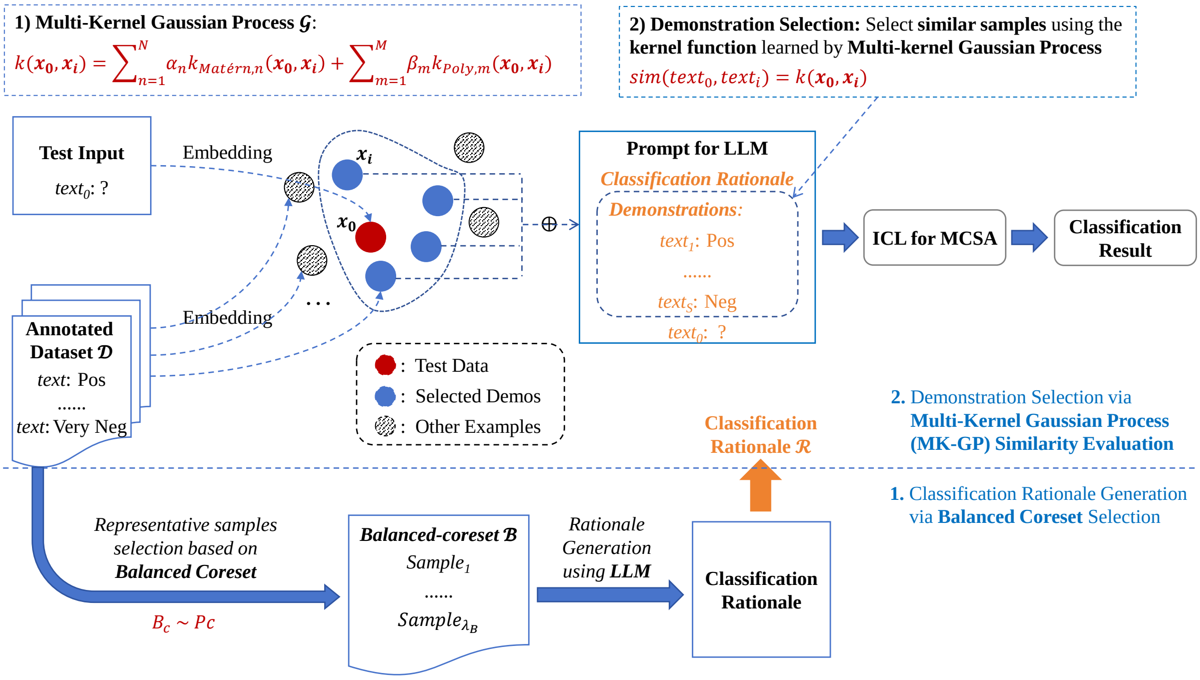

1) Multi-Kernel Gaussian Process $\mathcal{G}$:
$k(x_0, x_i) = \sum_{n=1}^{N} \alpha_n k_{Matérn,n}(x_0, x_i) + \sum_{m=1}^{M} \beta_m k_{Poly,m}(x_0, x_i)$
2) Demonstration Selection: Select similar samples using the kernel function learned by Multi-kernel Gaussian Process
$sim(text_0, text_i) = k(x_0, x_i)$
Test Input
$text_0$: ?
Embedding
x_i
x_0
Embedding
Annotated Dataset $\mathcal{D}$
text: Pos
......
text: Very Neg
Test Data
Selected Demos
Other Examples
Prompt for LLM
Classification Rationale
Demonstrations:
$text_1$: Pos
......
$text_S$: Neg
$text_0$: ?
ICL for MCSA
Classification Result
Classification Rationale $\mathcal{R}$
2. Demonstration Selection via Multi-Kernel Gaussian Process (MK-GP) Similarity Evaluation
1. Classification Rationale Generation via Balanced Coreset Selection
Representative samples selection based on Balanced Coreset
$B_c \sim Pc$
Balanced-coreset $\mathcal{B}$
$Sample_1$
......
$Sample_{\lambda_B}$
Rationale Generation using LLM
Classification Rationale

Figure 1: The framework of RD-MCSA: The lower half of the figure (below the long dashed line) corresponds to Section 3.1, while the upper half (above the long dashed line) corresponds to Section 3.2. The training of the MK-GP (described in Subsection 3.2.2) is omitted in the figure.

3 The Methodology of RD-MCSA

The RD-MCSA framework, illustrated in Fig. 1, consists of the following key components. Given an annotated MCSA dataset $\mathcal{D}$: 1) a balanced Coreset $\mathcal{B}$ is constructed to generate classification rationales $\mathcal{R}$ (Section 3.1); 2) a multi-kernel Gaussian process $\mathcal{G}$ is trained (Subsection 3.2.2) to model adaptive similarity; 3) for MCSA on a new input, ICL is performed using a prompt that incorporates both $\mathcal{R}$ and a set of demonstrations selected from $\mathcal{D}$ via $\mathcal{G}$ (Subsection 3.2.3).

3.1 Classification Rationale Generation via Balanced Coreset Selection

The classification rationales $\mathcal{R}$ are generated by an LLM through reasoning over the semantic and linguistic features of a representative subset of $\mathcal{D}$. To ensure that this subset (denoted as $\mathcal{B}$) preserves the semantic diversity and key distinguishing characteristics of each sentiment class—while also mitigating class imbalance—a **balanced Coreset selection algorithm** is proposed.

3.1.1 The Balanced Coreset Algorithm

The proposed algorithm extends the classical Coreset formulation (Sener and Savarese, 2017) by incorporating **importance-weighted sampling** and **class-aware stratification**, ensuring that the selected subset $\mathcal{B}$ maintains both intra-class diversity and inter-class balance, thereby facilitating higher-quality rationale generation.

To enforce class balance, the number of selected samples per class is capped by $\lambda'_{\mathcal{B}} = \left\lceil \frac{\lambda_{\mathcal{B}}}{u} \right\rceil$, where u denotes the number of unique sentiment classes in $\mathcal{D}$, and $\lambda_{\mathcal{B}}$ is a hyperparameter specifying the total Coreset size.

1) Importance-Weighted Sampling Probability. To prioritize semantically informative and potentially ambiguous instances, each sample is assigned a score based on its distance from the centroid of its respective class (Cohen-Addad et al., 2021).

For a given text sample $(t_i, y_i) \in \mathcal{D}$, let $\boldsymbol{x}(t_i) \in \mathbb{R}^d$ denote the embedding of t_i, where $y_i = c$ is its class label. The centroid $\boldsymbol{\mu}_c$ of class c is computed as $\boldsymbol{\mu}_c = \frac{1}{|\mathcal{D}_c|} \sum_{j:y_j=c} \boldsymbol{x}(t_j)$, where $\mathcal{D}_c \subset \mathcal{D}$ denotes the set of samples belonging to class c. The **importance weight** is defined as the squared Euclidean distance $w(t_i, y_i) = \|\boldsymbol{x}(t_i) - \boldsymbol{\mu}_c\|_2^2$.

Within each class, importance weights are normalized to form a probability distribution. The **sampling probability** of t_i, denoted as $P_c(t_i)$, is defined as:

$$P_c(t_i) = \frac{w(t_i, y_i)}{\sum_{j:y_j=c} w(t_j, y_j)}. \qquad (1)$$

2) Stratified Weighted Random Sampling. Sample selection is performed independently for each class $1 \leq c \leq u$, based on the corresponding sampling probabilities:

- If $|\mathcal{D}_c| \leq \lambda'_{\mathcal{B}}$, all instances from class c are included in $\mathcal{B}$.
- If $|\mathcal{D}_c| > \lambda'_{\mathcal{B}}$, a subset of $\lambda'_{\mathcal{B}}$ samples is drawn from $\mathcal{D}_c$ via weighted sampling with $P_c(t_i)$, forming the subset $\mathcal{B}_c$:

$$\mathcal{B}_c \subset \mathcal{D}_c, \quad |\mathcal{B}_c| = \lambda'_{\mathcal{B}}, \quad \mathcal{B}_c \sim P_c.$$

The final balanced Coreset $\mathcal{B}$ is obtained by aggregating all class-specific subsets $\mathcal{B}_c$.

3.1.2 Classification Rationale Generation via LLM Reasoning

To extract class-discriminative knowledge from the Coreset $\mathcal{B}$, classification rationales $\mathcal{R}$ are generated using an LLM guided by a carefully designed prompt. The use of LLMs for rationale generation leverages their advanced reasoning abilities (Wang, 2025), offering a scalable and semantically informed alternative to manual annotation.

In addition, since LLMs are later employed for ICL in downstream MCSA tasks, generating classification rationales with the same model family enhances alignment between rationale formulation and model interpretation.

> Based on the representative examples provided below, generate detailed descriptions for each sentiment label.
>
> **Examples:** {Balanced Coreset $\mathcal{B}$}
> **Sentiment Labels:** {*str(label_list)*}
>
> For each sentiment label, provide a comprehensive description covering:
>
> - **Lexical Patterns**
> - **Semantic-Pragmatic Features**
> - **Domain-Attribute Associations**

Figure 2: Prompt template for generating classification rationales using the balanced coreset $\mathcal{B}$.

The prompt instructs the LLM to identify key linguistic and semantic features that distinguish sentiment classes (as shown in Figure 2), focusing on: 1) **Lexical Patterns**: Characteristic sentiment-bearing words, phrases, and affective expressions; 2) **Semantic-Pragmatic Features**: Contextual meaning shifts and pragmatic implications across classes; 3) **Domain-Attribute Associations**: Domain-specific entities and properties linked to sentiment expression.

The LLM is further guided to ground its analysis in representative examples from $\mathcal{B}$, referencing specific lexical or syntactic patterns. This ensures the resulting rationales are both interpretable and empirically supported.

3.2 Demonstration Selection via Multi-Kernel Gaussian Process Similarity Evaluation

RD-MCSA leverages a **multi-kernel Gaussian process for text similarity evaluation** to select ICL demonstrations. This method benefits from Multiple Kernel Learning's ability to model and adapt to complex data distributions (Ghasempour and Martínez-Ramón, 2023).

3.2.1 Gaussian Process

Gaussian Process (GP) (Liu et al., 2021) can be applied to model categorical data with u categories by introducing a set of latent functions $\{f_c(\boldsymbol{x})\}_{c=1}^{u}$, one for each class. Each latent function is modeled as an independent Gaussian Process (Wang, 2023):

$$f_c(\boldsymbol{x}) \sim \mathcal{GP}(e_c(\boldsymbol{x}), k_c(\boldsymbol{x}, \boldsymbol{x'})), \tag{2}$$

where $e_c(\boldsymbol{x})$ denotes the mean function, and $k_c(\boldsymbol{x}, \boldsymbol{x'})$ represents the covariance function (also referred to as the **kernel**) for the c-th class.

Following prior work such as (Bonilla et al., 2007), this study adopts a shared kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and a shared mean function across all categories. This design choice not only reduces computational complexity but also capitalizes on structural similarities commonly observed among different classes within the same dataset. In this framework, the mean function is modeled as a learnable constant, and the kernel is defined as a multi-kernel function, as described in Section 3.2.2.

3.2.2 Multi-Kernel Gaussian Process

Multi-Kernel Gaussian Process (MK-GP) extends the standard Gaussian Process by integrating Multiple Kernel Learning. A weighted combination of the Matérn kernel (Borovitskiy et al., 2021) and the polynomial kernel (Song et al., 2021) is employed, enabling the model to **effectively characterize both stationary and non-stationary behaviors in the data** (Lawler, 2018). The combined kernel function is defined as follows:

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{n=1}^{N} \alpha_n k_{\text{Matérn},n}(\boldsymbol{x_i}, \boldsymbol{x_j}) + \sum_{m=1}^{M} \beta_m k_{Poly,m}(\boldsymbol{x_i}, \boldsymbol{x_j}), \tag{3}$$

where $k_{\text{Matérn},n}(\boldsymbol{x_i}, \boldsymbol{x_j})$ denotes the n-th Matérn kernel, and $k_{\text{Poly},m}(\boldsymbol{x_i}, \boldsymbol{x_j})$ denotes the m-th polynomial kernel. The coefficients α_n and β_m are learnable weights constrained to be non-negative ($\alpha_n, \beta_m \geq 0$). Additional details are provided in Appendix A.

Let $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{K}$ denote the training data and $\boldsymbol{y}$ represent the corresponding labels. Let $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_u(\boldsymbol{x})]^T$ denote the vector of latent function values at input $\boldsymbol{x}$, and let $\boldsymbol{f} = \{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^{K}$ denote the collection of latent outputs over the training set. An MK-GP model $\mathcal{G}$ is trained by minimizing the loss function, which is the **negative log-marginal likelihood** (Artemev et al., 2021):

$$\mathcal{L} = -\log \int p(\boldsymbol{y} \mid \boldsymbol{f})\, p(\boldsymbol{f} \mid \boldsymbol{X})\, d\boldsymbol{f}. \tag{4}$$

3.2.3 Similarity-Based Demonstration Selection via the Kernel Function

Similarity-based demonstration selection, which selects examples most similar to the test sample, has proven effective for ICL (Margatina et al., 2023). In this work, we adopt a similarity-based approach leveraging the kernel function of the trained MK-GP model $\mathcal{G}$ to guide demonstration selection. Given a test sample t_0, its similarity to a candidate example $t_i \in \mathcal{D}$ is computed as:

$$sim(t_0, t_i) = k(\boldsymbol{x}(t_0), \boldsymbol{x}(t_i)), \tag{5}$$

where $\boldsymbol{x}(t_0)$ and $\boldsymbol{x}(t_i)$ (or, for brevity, $\boldsymbol{x}_0$ and $\boldsymbol{x}_i$) are the embeddings of t_0 and t_i, respectively. As shown in Figure 3, the embeddings are mapped into a Hilbert space via a kernel function. With a well-chosen kernel, the transformed representations exhibit improved class separability relative to the original embedding space (Elen et al., 2022). This enhanced structure enables more discriminative similarity computation for ICL. A higher kernel value (as learned in Section 3.2.2) reflects greater similarity between examples in the feature space (Thickstun, 2019). Additional implementation details are provided in Appendix B.

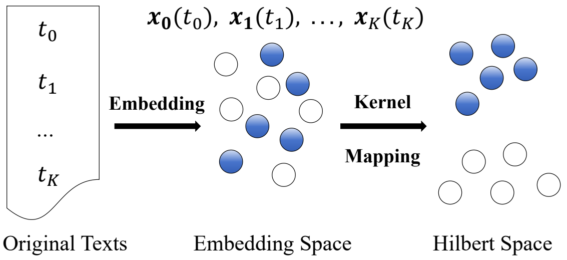


Figure 3: Kernel mapping enhances class separability. Circles in two different colors represent samples from distinct classes.

The S examples most similar to t_0 are selected as demonstration examples. These examples, along with their corresponding labels, denoted as $\{(t_1, y_1), \ldots, (t_S, y_S)\}$, are then concatenated with the classification rationale $\mathcal{R}$ to form a 'prompt' (as shown in Figure 4) for the LLM. This process is defined as follows:

$$\hat{y_0} = \text{LLM}(t_0 \oplus \mathcal{R} \oplus (t_1, y_1) \oplus \cdots \oplus (t_S, y_S)),$$

where $\hat{y_0}$ is the predicted label for t_0, and $\oplus$ represents the concatenation operation.

> Analyze the sentiment expressed in the given **Query Text** toward the specified target {*target*}. The sentiment label must be selected from the following set: {*str(label_list)*}. Refer to the provided label descriptions and example demonstrations to guide your classification.
>
> **Label Descriptions:** {Rationales $\mathcal{R}$}
> **Demonstrations:** $\{(t_1, y_1), \ldots, (t_S, y_S)\}$
>
> **Query Text:** {*query_text*}

Figure 4: Prompt template of ICL for MCSA.

4 Experimental Setup

4.1 Experimental Datasets

To evaluate RD-MCSA, experiments were conducted on five diverse datasets across various domains and sentiment classification granularities as shown in Table 1:

Dataset	Size	Classes	Granularity & Text type
SST5[1]	11,855	5	Sentence-level Movie Reviews
SemEval17[2]	20,632	5	Topic-based Tweets
ABSIA[3]	4,650	7	Restaurant-related Reviews
PR_Baby[4]	183,531	5	Baby-product Reviews
PR_Software[5]	12,804	5	Software Product Reviews

Table 1: Summary of experimental datasets.

These datasets cover a range of sentiment classification tasks, from sentence-level analysis to fine-grained aspect-based sentiment analysis, enabling a comprehensive evaluation of RD-MCSA.

4.2 Experimental Implementation Details

In the experiments, 1,000 instances were randomly sampled from each dataset to construct the annotated dataset $\mathcal{D}$, ensuring a fair evaluation of RD-MCSA across datasets. This also provided insights into the amount of labeled data required for MCSA tasks, aiding in determining the annotation needed to outperform traditional classifiers trained on large-scale datasets. The balanced Coreset size for generating the classification rationale was set to $\lambda_{\mathcal{B}} = 100$. Taking into account both efficiency and effectiveness, the number of demonstrations was set to $S = 10$.

Experiments were conducted using three groups of LLMs: GPT[6], DeepSeek[7], and ERNIE[8]. For each group, the more capable (and expensive) model (GPT-4o, DeepSeek-R1, and ERNIE X1 Turbo) was employed for classification rationale generation, whereas the more cost-efficient variant (GPT-4o-mini, DeepSeek-V3, and ERNIE 4.5 Turbo) was utilized for ICL in MCSA tasks.

The following settings were applied uniformly across all datasets: $N = 9$ and $M = 9$ were used in the MK-GP model (Equation (3)). The Adam optimizer was adopted with a learning rate of 0.01 over 500 training epochs, and all other optimizer parameters were set to their default values. Optimal hyperparameters were selected via grid search and cross-validation.

Most experiments were conducted on an NVIDIA GeForce RTX 3080 GPU. On average, a single unit of this GPU required 170.86 seconds to complete 500 epochs of Gaussian process training across various datasets. For API-based models, remote inference was employed instead.

4.3 Comparison Models

Baseline models were selected from two categories: (1) classic machine learning and (2) language models for sentiment classification. The selected models were: 1) **Naïve Bayes** (Rennie, 2001): Multinomial Naïve Bayes with TF-IDF features, using class weighting to address class imbalance. 2) **SVM** (Li et al., 2011): Support Vector Classifier with a linear kernel, balanced class weights, and TF-IDF features. 3) **BERT** (Sun et al., 2019): BERT-base model fine-tuned with Focal Loss to mitigate class imbalance. 4) **BERTweet** (Nguyen et al., 2020): Pretrained model for English tweets, also optimized with Focal Loss to address imbalance.

All baseline models were trained and evaluated on the datasets using an 80%/20% train-test split.

Given the recent success of ICL approaches in text classification, several ICL-based selection strategies were included as **comparison methods**: 1) **Random**: Selected in-context examples randomly from the candidate set. 2) **Coreset** (Indyk et al., 2014): Selected representative samples that reflect overall dataset diversity. 3) **Cos-Similarity** (de Vos et al., 2022): Selected the top-S examples based on cosine similarity. 4) **BM25** (Robertson et al., 2009): Selected the top-S examples using BM25 scoring. 5) **Complex-CoT** (Fu et al., 2022): Selected examples based on complexity, measured via the number of newline characters. 6) **Auto-CoT** (Zhang et al., 2022b): Clustered candidate examples and selected those closest to each cluster center.

To ensure a fair comparison, all ICL-based methods were applied to the same annotated dataset of 1,000 labeled samples as RD-MCSA, with 100 demonstrations ($S = 100$). In addition, all prompts incorporated classification rationales generated by the same method.

4.4 Evaluation Metric

Due to the multi-class nature of MCSA and the class imbalance in the experimental data, Accuracy and weighted-average F1 score were used to evaluate performance (Sokolova and Lapalme, 2009).

5 Experimental Results and Analysis

5.1 Main Results

Table 2 summarizes the performance of various methods on three datasets. The results for the remaining two datasets are presented in Appendix C. The following observations can be made:

1) Effectiveness of ICL. ICL achieved the highest Accuracy and weighted F1 scores across all datasets, outperforming both traditional machine learning models and language model clas-

[1] https://huggingface.co/datasets/SetFit/sst5

[2] https://huggingface.co/datasets/midas/semeval2017

[3] https://www.iitp.ac.in/ãi-nlp-ml/resources.html#ABSIA

[4] https://snap.stanford.edu/data/web-Amazon-links.html

[5] https://cseweb.ucsd.edu/˜jmcauley/datasets/amazon_v2

[6] https://openai.com/api/

[7] https://www.deepseek.com/

[8] https://yiyan.baidu.com/

Table 2: Experimental results of baseline methods and ICL approaches across three datasets, using three groups of LLMs. The best-performing method within each category is highlighted in bold.

	Method	**SST5**		**SemEval17**		**ABSIA**	
		Acc (%)	F1 (%)	Acc (%)	F1 (%)	Acc (%)	F1 (%)
Baseline Models	Naïve Bayes	37.2	37.0	44.9	44.0	34.8	31.0
	SVM	37.1	37.0	56.7	58.0	49.9	50.0
	BERT	**49.9**	**50.0**	59.2	61.0	51.2	**52.0**
	BERTweet	48.7	47.0	**63.4**	**65.0**	**52.4**	**52.0**
ICL based on GPT-4o +GPT-4o-mini	Random	55.0	54.90	57.7	60.22	51.6	52.87
	Coreset	55.7	55.44	59.4	62.07	53.2	55.39
	Cos-Similarity	55.6	55.08	60.1	61.92	52.8	53.58
	BM25	56.5	56.02	61.6	63.53	53.0	54.66
	Complex-CoT	56.5	54.30	62.5	63.12	52.9	55.26
	Auto-CoT	56.6	54.18	62.2	63.09	53.4	55.62
	RD-MCSA	**57.6**	**56.03**	**63.9**	**64.69**	**54.3**	**56.01**
ICL based on DeepSeek-R1 +DeepSeek-V3	Random	56.1	55.18	67.2	67.71	51.2	53.26
	Coreset	56.2	55.09	67.6	68.4	52.7	53.98
	Cos-Similarity	56.3	55.21	68.4	**68.62**	53.2	55.41
	BM25	56.6	55.75	67.3	67.99	53.1	54.72
	Complex-CoT	56.1	53.84	67.5	67.31	52.2	53.36
	Auto-CoT	56.3	54.64	67.7	68.11	52.7	54.99
	RD-MCSA	**57.9**	**57.00**	**68.6**	68.55	**54.6**	**56.50**
ICL based on ERNIE X1 Turbo +ERNIE 4.5 Turbo	Random	51.3	48.80	67.2	66.91	50.5	50.59
	Coreset	53.3	52.21	67.4	66.97	51.2	52.23
	Cos-Similarity	55.1	53.26	67.5	67.00	52.9	52.21
	BM25	54.7	53.18	67.7	67.14	52.8	52.47
	Complex-CoT	56.1	53.74	67.9	67.27	52.1	52.36
	Auto-CoT	52.2	51.47	67.6	67.21	52.7	52.51
	RD-MCSA	**57.1**	**55.99**	**69.1**	**68.31**	**53.4**	**53.46**

sifiers. Remarkably, ICL used only 1,000 labeled examples—substantially fewer than the tens of thousands required by the baseline methods—demonstrating both superior efficiency and effectiveness.

2) Effectiveness of RD-MCSA: RD-MCSA consistently outperformed other methods on most datasets, with the exception of SemEval17, where the cosine similarity-based ICL method achieved a slightly higher F1 score. These results underscore the robustness and effectiveness of RD-MCSA, further corroborated by additional ablation studies.

3) Comparison of Demonstration Selection Methods: Structured demonstration selection strategies, such as Coreset, Auto-CoT, and similarity-based approaches including BM25, Cosine, and RD-MCSA, consistently outperformed random sampling. Among these methods, RD-MCS demonstrated the highest effectiveness in identifying informative examples for ICL.

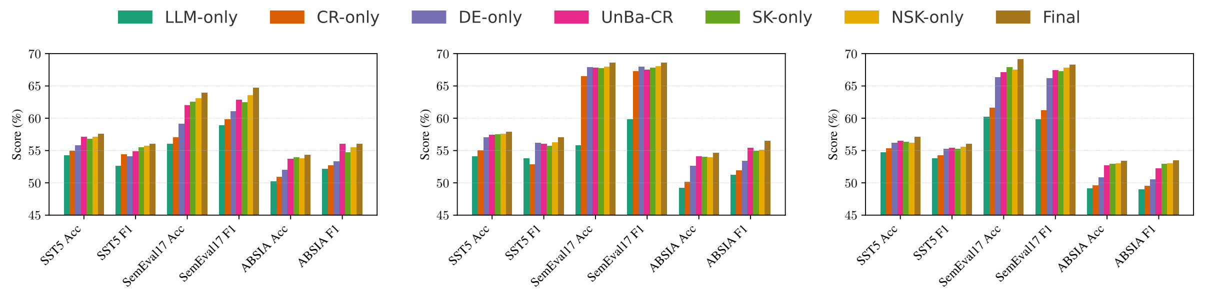
5.2 Ablation Analysis

For further analysis, **ablation studies** were conducted with the following model variants: 1) **LLM-only**: Relied solely on the LLM's inherent reasoning for classification, without classification rationales or demonstration examples. 2) **CR-only**: Used only classification rationales in the prompt, excluding demonstration examples. 3) **DE-only**: Used only demonstration examples, excluding classification rationales. 4) **UnBa-CR**: Omitted category balancing when generating classification rationales. 5) **SK-only**: Employed only stationary kernel functions in the MK-GP algorithm. 6) **NSK-only**: Employed only non-stationary kernel functions in the MK-GP algorithm.

Figure 5 presents the results of the ablation study conducted on three datasets (results for the remaining two datasets are provided in Appendix D). The following conclusions can be drawn:

1) Effectiveness of Rationales: Incorporating classification rationales led to improved performance compared to direct classification. Rationales enhanced the LLM's ability to interpret label meanings, thereby improving classification accuracy.

2) Effectiveness of Demonstrations: Including demonstration examples significantly boosted performance compared to direct classification. These



(a) GPT-based (b) DeepSeek-based (c) ERNIE-based

Figure 5: Experimental results from ablation studies across all datasets demonstrate that the removal of any component from the RD-MCSA algorithm leads to a measurable decline in performance.

Table 3: Per-sample average inference time (in seconds) of various ICL methods on three datasets.

Backbone	Method	SST5	SemEval17	ABSIA
ICL based on GPT-4o +GPT-4o-mini	Random	8.72	9.05	7.58
	Coreset	8.83	8.93	7.62
	Cos-Similarity	8.86	9.15	7.72
	BM25	9.02	9.22	7.81
	Complex-CoT	8.74	9.13	7.25
	Auto-CoT	8.81	9.21	7.43
	RD-MCSA	8.91	9.17	7.73
ICL based on DeepSeek-R1 +DeepSeek-V3	Random	12.90	12.61	7.78
	Coreset	13.14	13.34	8.42
	Cos-Similarity	13.50	13.20	8.23
	BM25	13.21	13.64	8.61
	Complex-CoT	12.97	12.81	8.25
	Auto-CoT	13.12	12.78	8.11
	RD-MCSA	13.17	13.82	8.57
ICL based on ERNIE X1 Turbo +ERNIE 4.5 Turbo	Random	10.98	11.01	7.66
	Coreset	11.07	11.21	7.79
	Cos-Similarity	11.82	11.17	8.21
	BM25	11.23	11.61	8.33
	Complex-CoT	11.19	11.32	8.91
	Auto-CoT	11.11	11.49	7.98
	RD-MCSA	11.36	11.44	8.11

demonstrations served as concrete references that guided the LLM's decision-making process.

3) Impact of Label Imbalance in Rationale Generation: Generating classification rationales from imbalanced training samples resulted in noticeable performance degradation. The scarcity of examples from minority classes impaired the LLM's ability to generalize and reduced the quality of the generated rationales.

4) Effectiveness of Combined Stationary and Non-Stationary Kernels: Combining stationary and non-stationary kernels resulted in better performance than using either type alone. This combination more effectively captured structural complexity and enabled the selection of more similar examples in ICL, thereby improving classification accuracy.

5.3 Time Cost Analysis

The computational overhead of RD-MCSA comprises two main components: (1) the *offline stage*, which involves Coreset pool construction, rationale generation, and MK-GP training; and (2) the *ICL inference stage*. Statistical analysis based on Table 3, which reports the per-sample average inference time of various ICL methods across three datasets (with results for additional datasets provided in Table 5, Appendix E), indicates that there is no statistically significant difference in inference-time cost among the evaluated algorithms. Detailed results are presented in Appendix E. Therefore, the additional computational overhead introduced by RD-MCSA is limited to the offline preprocessing stage.

6 Conclusions

This paper presents a novel framework for multi-class sentiment analysis (MCSA) that leverages in-context learning (ICL) by integrating classification rationale generation based on balanced Coreset sampling and demonstration selection using multi-kernel Gaussian processes (MK-GP). The proposed approach effectively addresses key challenges such as class imbalance and the high cost of large-scale annotation, while also capturing subtle and nuanced sentiment expressions.

Extensive experiments across five diverse datasets demonstrate the superior performance, robustness, and generalizability of the method.

Future research directions include extending the framework to other sentiment analysis tasks, incorporating multimodal data (e.g., audio and visual inputs), improving computational efficiency, and designing strategies to mitigate the effects of subjectivity in annotation. These advancements are expected to further contribute to the development of more accurate, efficient, and scalable sentiment analysis systems.

Limitations

This paper has the following limitations:

1. Although the proposed method has been validated on five diverse datasets, its applicability remains somewhat limited. In particular, it has not yet been evaluated on multimodal datasets, which are increasingly important in real-world scenarios.

2. The overall performance of the proposed method, while promising, is still not sufficiently high. Even traditional supervised models trained on tens of thousands of samples often struggle to surpass 80% accuracy. A major challenge in MCSA tasks stems from the inherent subjectivity of annotations—different annotators may assign different labels to the same sample, thus limiting classification performance. Additionally, the quality of benchmark datasets may vary, and a thorough analysis of this factor has not been conducted.

3. Although the MK-GP approach demonstrates strong results, it is computationally more intensive than some similarity evaluation methods, especially in the offline stage. Enhancing its computational efficiency represents an important avenue for future research that remains unexplored in the current work.

Ethics Statement

Our study uses publicly available datasets, and no personally identifiable information is included. We acknowledge potential biases in sentiment classification tasks and have taken steps to mitigate them, such as dataset balancing and bias analysis. No human subjects were involved in the study, and no additional ethical approval was required. While our method could be used for sentiment analysis applications, we do not foresee direct misuse. We will release the code and models responsibly, ensuring compliance with ethical guidelines.

LLMs (mainly GPT) are applied in our writing to help correct grammatical and word usage errors, but they do not generate any ideas, data, images, or tables for us.

References

Arwa SM AlQahtani. 2021. Product sentiment analysis for amazon reviews. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, 13.

Artem Artemev, David R Burt, and Mark van der Wilk. 2021. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In *International Conference on Machine Learning*, pages 362–372. PMLR.

Edwin V Bonilla, Kian Chai, and Christopher Williams. 2007. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20.

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. 2021. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR.

Yunmo Chen, Tongfei Chen, Harsh Jhamtani, Patrick Xia, Richard Shin, Jason Eisner, and Benjamin Van Durme. 2024. Learning to retrieve iteratively for in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7156–7168, Miami, Florida, USA. Association for Computational Linguistics.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy Rogers. 2024. Simulating opinion dynamics with networks of LLM-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3326–3346, Mexico City, Mexico. Association for Computational Linguistics.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. 2021. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 169–182.

Lynne M Connelly. 2021. Introduction to analysis of variance (anova). *Medsurg Nursing*, 30(3).

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 519–535.

Dayan de França Costa and Nadia Felix Felipe da Silva. 2018. Inf-ufg at fiqa 2018 task 1: predicting sentiments and aspects on financial tweets and news headlines. In *Companion Proceedings of the The Web Conference 2018*, pages 1967–1971.

Isa M Apallius de Vos, Ghislaine L Boogerd, Mara D Fennema, and Adriana D Correia. 2022. Comparing in context: Improving cosine similarity measures with a metric tensor. *arXiv preprint arXiv:2203.14996*.

Abdullah Elen, Selçuk Baş, and Cemil Közkurt. 2022. An adaptive gaussian kernel for support vector machine. *Arabian Journal for Science and Engineering*, 47(8):10579–10588.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Alireza Ghasempour and Manel Martínez-Ramón. 2023. Multiple output sparse gaussian processes with multiple kernel learning for electric load forecasting. In *2023 5th International Conference on Power and Energy Technology (ICPET)*, pages 987–990. IEEE.

Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S Mirrokni. 2014. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 100–108.

Lakshmi Revathi Krosuri and Rama Satish Aravapalli. 2023. Novel heuristic-based hybrid resnext with recurrent neural network to handle multi class classification of sentiment analysis. *Machine Learning: Science and Technology*, 4(1):015033.

Gregory F Lawler. 2018. *Introduction to stochastic processes*. Chapman and Hall/CRC.

Lung-Hao Lee, Liang-Chih Yu, Suge Wang, and Jian Liao. 2024. Overview of the sighan 2024 shared task for chinese dimensional aspect-based sentiment analysis. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 165–174.

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 (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Kunlun Li, Jing Xie, Xue Sun, Yinghui Ma, and Hui Bai. 2011. Multi-class text categorization based on lda and svm. *Procedia Engineering*, 15:1963–1967.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Haitao Liu, Yew-Soon Ong, Ziwei Yu, Jianfei Cai, and Xiaobo Shen. 2021. Scalable gaussian process classification with additive noise for non-gaussian likelihoods. *IEEE transactions on cybernetics*, 52(7):5842–5854.

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.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Mamta and Asif Ekbal. 2023. Service is good, very good or excellent? towards aspect based sentiment intensity analysis. In *European Conference on Information Retrieval*, pages 685–700. Springer.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Emilio Porcu, Moreno Bevilacqua, Robert Schaback, and Chris J Oates. 2024. The matérn model: A journey through statistics, numerical analysis and machine learning. *Statistical Science*, 39(3):469–492.

Korbinian Randl, John Pavlopoulos, Aron Henriksson, and Tony Lindgren. 2024. CICLe: Conformal in-context learning for largescale multi-class food risk classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7695–7715, Bangkok, Thailand. Association for Computational Linguistics.

Jason DM Rennie. 2001. Improving multi-class text classification with naive bayes.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019. Semeval-2017 task 4: Sentiment analysis in twitter. *arXiv preprint arXiv:1912.00741*.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Neeraj Anand Sharma, ABM Shawkat Ali, and Muhammad Ashad Kabir. 2024. A review of sentiment analysis: tasks, applications, and deep learning techniques. *International journal of data science and analytics*, pages 1–38.

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 *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5248–5259, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437.

Zhao Song, David Woodruff, Zheng Yu, and Lichen Zhang. 2021. Fast sketching of polynomial kernels of polynomial degree. In *International Conference on Machine Learning*, pages 9812–9823. PMLR.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer.

John Thickstun. 2019. Mercer's theorem. *University of Washington, dostupné na internete (5.2. 2018): https://homes. cs. washington. edu/˜ thickstn/docs/mercer. pdf.*

Jie Wang. 2023. An intuitive tutorial to gaussian processes regression. *Computing in Science & Engineering*.

Jun Wang. 2025. A tutorial on llm reasoning: Relevant methods behind chatgpt o1. *Preprint*, arXiv:2502.10867.

Zhaoxia Wang, Zhenda Hu, Seng-Beng Ho, Erik Cambria, and Ah-Hwee Tan. 2023. Mimusa—mimicking human language understanding for fine-grained multi-class sentiment analysis. *Neural Computing and Applications*, 35(21):15907–15921.

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.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico. Association for Computational Linguistics.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022a. 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.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

A Properties of Kernel Functions

The polynomial kernel is expressed as:

$$k_{Poly,m}(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\gamma_m \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle + c_m)^{d_m} ,$$

where γ_m is a scaling factor, c_m is an offset (both learnable parameters), and d_m is the degree of the polynomial, treated as a hyper-parameter. Here, $\langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle$ denotes the dot product of $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$.

The Matérn kernel is defined as follows, where ν and ℓ are the kernel parameters:

$$k_{\text{Matérn}}(x_i, x_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|x_i - x_j\|}{\ell} \right)^{\nu} B_\nu \left(\sqrt{2\nu} \frac{\|x_i - x_j\|}{\ell} \right),$$

where $\Gamma(\nu)$ represents the Gamma function, defined as:

$$\Gamma(\nu) = \int_0^\infty t^{\nu-1} e^{-t} \, dt,$$

Here, $B_\nu(z)$ denotes the modified Bessel function of the second kind, defined as:

$$B_\nu(z) = \frac{\pi}{2} \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}.$$

where $I_\nu(z)$ is the modified Bessel function of the first kind, given by:

$$I_\nu(z) = \sum_{k=0}^{\infty} \frac{\left(\frac{z}{2}\right)^{\nu+2k}}{k!\Gamma(\nu+k+1)},$$

The following analysis reveals the limiting behavior of the Matérn kernel, which approaches two commonly used stationary kernels—namely, the RBF kernel and the Laplace kernel—under different conditions. This serves as the motivation for employing the Matérn kernel in this paper to characterize stationarity.

Table 4: Experimental results of baseline methods and ICL approaches using three groups of LLMs on two datasets, with the best-performing method in each category shown in bold.

	Method	PR_Baby		PR_Software	
		Acc (%)	F1 (%)	Acc (%)	F1 (%)
Baseline Models	Naïve Bayes	47.86	47.0	44.8	45.0
	SVM	50.96	51.0	58.1	59.0
	BERT	**58.18**	**58.0**	**60.3**	**61.0**
	BERTweet	57.74	56.0	59.9	58.0
ICL based on GPT-4o +GPT-4o-mini	Random	57.9	57.88	62.3	63.57
	Coreset	58.1	58.06	62.6	63.68
	Cos-Similarity	58.9	59.03	64.7	65.86
	BM25	59.2	59.36	63.1	64.25
	Complex-CoT	58.4	58.46	65.3	66.38
	Auto-CoT	58.8	59.07	62.7	64.08
	RD-MCSA	**60.1**	**60.32**	**67.0**	**67.22**
ICL based on DeepSeek-R1 +DeepSeek-V3	Random	56.0	56.13	61.5	62.94
	Coreset	56.3	56.42	63.5	64.54
	Cos-Similarity	56.6	56.72	64.5	65.91
	BM25	56.6	56.74	63.9	65.09
	Complex-CoT	56.4	56.58	65.7	65.29
	Auto-CoT	56.5	56.64	63.2	64.49
	RD-MCSA	**57.5**	**57.70**	**67.7**	**68.11**
ICL based on ERNIE X1 turbo +ERNIE 4.5 turbo	Random	55.8	55.13	62.7	62.34
	Coreset	56.0	56.47	64.1	64.13
	Cos-Similarity	56.6	56.65	64.6	65.07
	BM25	56.9	56.21	64.7	64.83
	Complex-CoT	56.2	56.33	65.5	65.17
	Auto-CoT	56.7	56.53	66.0	66.21
	RD-MCSA	**57.8**	**56.88**	**66.5**	**67.47**

When the parameter $\nu \to \infty$, the Matérn kernel converges to the Radial Basis Function (RBF) kernel (Porcu et al., 2024):

$$\lim_{\nu \to \infty} k_{\text{Matérn}}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\ell^2}\right).$$

When the parameter $\nu = \frac{1}{2}$, the Matérn kernel becomes equivalent to the Laplace kernel [54]:

$$k_{\text{Matérn}}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\ell}\right), \text{when } \nu = \frac{1}{2}.$$

B Similarity Evaluation Based on Kernel Functions of MK-GP

According to Mercer's theorem (Thickstun, 2019), there exists a Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ such that the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ can be expressed as the inner product in the Hilbert space:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}, \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}.$$

Here, $\phi(\mathbf{x})$ is an implicitly defined mapping, and $\mathcal{H}$ is the corresponding Hilbert space. In $\mathcal{H}$, the Euclidean distance between any two samples $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ is defined as:

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle_{\mathcal{H}} - 2\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}} + \langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}.$$

By utilizing the definition of the kernel function, $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$, the expression can be rewritten as:

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j).$$

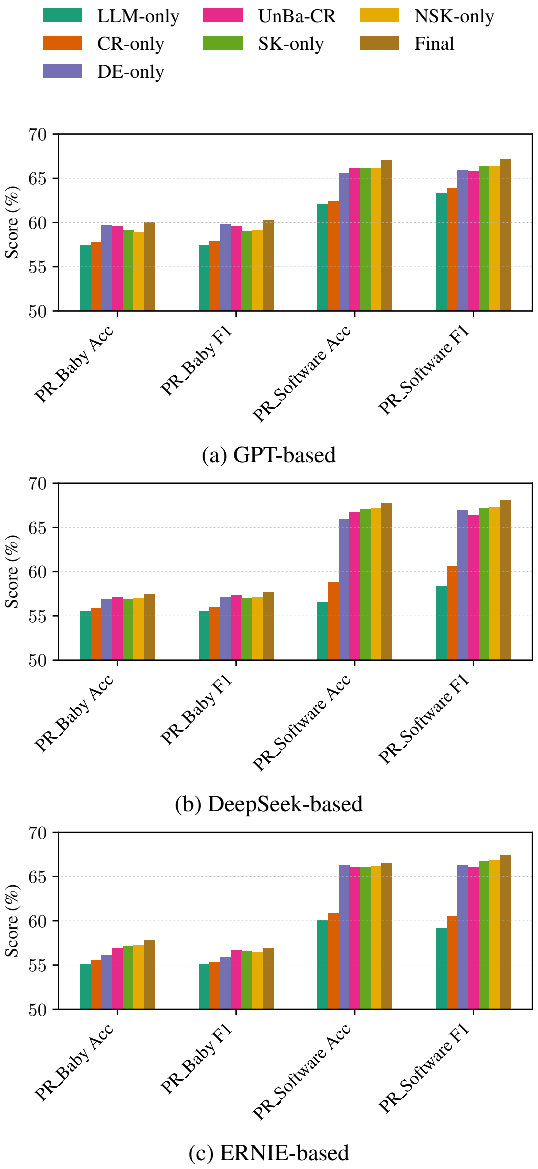
This represents the distance in the Hilbert space induced by a positive definite kernel function. After normalizing the samples, for the kernel function adopted in this study, the first and third terms in the above equation become constants. Thus, the larger the value of the middle term $k(\mathbf{x}_i, \mathbf{x}_j)$, the smaller the distance between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$, indicating that the two samples are more similar.

C Experimental Results on Other Two Datasets

Table 4 presents the experimental results of baseline models and ICL comparison models on the remaining two datasets.

D Ablation Study on Other Datasets

Figure 6 presents the experimental results of ablation study on the remaining two datasets.



(a) GPT-based

(b) DeepSeek-based

(c) ERNIE-based

Figure 6: Experimental results of ablation studies on two datasets. The removal of any component from the RD-MCSA algorithm resulted in a performance degradation.

E Time Cost Analysis Based on Variance Analysis

Table 5: Per-sample average inference time (in seconds) of various ICL methods on two datasets.

Backbone	Method	PR_Baby	PR_Software
ICL based on GPT-4o +GPT-4o-mini	Random	9.01	9.79
	Coreset	9.22	9.13
	Cos-Similarity	7.95	9.71
	BM25	9.31	7.72
	Complex-CoT	8.83	9.18
	Auto-CoT	7.76	10.12
	RD-MCSA	8.97	9.54
ICL based on DeepSeek-R1 +DeepSeek-V3	Random	11.36	12.19
	Coreset	12.31	12.88
	Cos-Similarity	13.31	14.12
	BM25	13.48	13.79
	Complex-CoT	11.21	13.11
	Auto-CoT	11.17	12.99
	RD-MCSA	12.21	12.92
ICL based on ERNIE X1 Turbo +ERNIE 4.5 Turbo	Random	10.59	11.98
	Coreset	11.32	12.17
	Cos-Similarity	10.27	11.82
	BM25	12.21	12.55
	Complex-CoT	11.64	12.88
	Auto-CoT	11.71	11.76
	RD-MCSA	12.55	11.72

Let $a = 7$, $b = 3$, and $c = 5$ respectively denote the number of levels for the three factors: *Algorithm* ($i = 1, \dots, a$), *Model* ($j = 1, \dots, b$), and *Dataset* ($k = 1, \dots, c$). For each combination of these factors, the inference time Y_{ijk} is recorded, and the variability is analyzed using a main-effects ANOVA model(Connelly, 2021), which assumes additive and independent factor effects without interactions:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \varepsilon_{ijk},$$

where the error terms are assumed to be independent and normally distributed with constant variance: $\varepsilon_{ijk} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$.

To assess whether the choice of algorithm significantly affects inference time, the following hypothesis test is conducted for the *Algorithm* factor:

$$H_0 : \alpha_1 = \cdots = \alpha_7 \quad \text{vs} \quad H_1 : \exists\, \alpha_i \neq \alpha_{i'},$$

The Sum of Squares for Factor A (SSA) is computed based on the deviation of each algorithm's marginal mean from the overall grand mean. Since each algorithm is evaluated across $b = 3$ models and $c = 5$ datasets, the number of replications is $bc = 15$:

$$\text{SSA} = \sum_{i=1}^{a} bc\,(\bar{Y}_{i..} - \bar{Y}_{...})^2 = 0.171,$$

The Sum of Squares for Error (SSE) is obtained by aggregating the squared deviations of each observation from its corresponding algorithm-level mean:

$$\text{SSE} = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{c} (Y_{ijk} - \bar{Y}_{i..})^2 = 2.793,$$

The degrees of freedom for factor A and the residual error term are:

$$\mathrm{df}_A = a - 1 = 6,$$

$$\mathrm{df}_E = abc - a - b - c + 2 = 92,$$

The corresponding mean squares are computed as:

$$\text{MSA} = \frac{\text{SSA}}{\mathrm{df}_A} = \frac{0.171}{6} = 0.0285,$$

$$\text{MSE} = \frac{\text{SSE}}{\mathrm{df}_E} = \frac{2.793}{92} = 0.0303,$$

The F-statistic for testing the algorithm factor is then given by:

$$F_{\text{obs}} = \frac{\text{MSA}}{\text{MSE}} = \frac{0.0285}{0.0303} = 0.94.$$

From the F-distribution table, the critical value at the 5% significance level is $F_{0.95}(6, 92) \approx 2.19$. Since $F_{\text{obs}} = 0.94 < 2.19$ ($p = 0.471 > 0.05$), the null hypothesis cannot be rejected. This indicates that, at the 0.05 significance level, the seven demonstration-selection algorithms do **not** exhibit statistically significant differences in per-sample inference time after accounting for the effects of model and dataset.