

# Dual Debiasing for Noisy In-Context Learning for Text Generation

**Siqi Liang**

University of Michigan  
Ann Arbor, MI, USA  
siqilian@umich.edu

**Paramveer S. Dhillon**

University of Michigan  
Ann Arbor, MI, USA  
dhillonp@umich.edu

**Sumyeong Ahn**

KENTECH  
Naju, Jeollanam-do, Korea  
sumyeongahn@kentech.ac.kr

**Jiayu Zhou\***

University of Michigan  
Ann Arbor, MI, USA  
jiayuz@umich.edu

## Abstract

In-context learning (ICL) relies heavily on high-quality demonstrations drawn from large annotated corpora. Existing approaches detect noisy annotations by ranking local perplexities, presuming that noisy samples yield higher perplexities than their clean counterparts. However, this assumption breaks down when the noise ratio is high and many demonstrations are flawed. We re-examine the perplexity-based paradigm for text generation under noisy annotations, highlighting two sources of bias in perplexity: the annotation itself and the domain-specific knowledge inherent in large language models (LLMs). To overcome these biases, we introduce a dual-debiasing framework that uses synthesized neighbors to explicitly correct perplexity estimates, yielding a robust *Sample Cleanliness Score*. This metric uncovers absolute sample cleanliness regardless of the overall corpus noise level. Extensive experiments demonstrate our method’s superior noise-detection capabilities and show that its final ICL performance is comparable to that of a fully clean demonstration corpus. Moreover, our approach remains robust even when noise ratios are extremely high.

## 1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities across a wide range of Natural Language Processing (NLP) tasks (Brown et al., 2020; Touvron et al., 2023). This performance is largely attributed to various techniques that leverage LLMs, such as Chain-of-Thought (CoT) (Wei et al., 2022), In-Context Learning (ICL) (Dong et al., 2024), and so on. In particular, ICL guides LLMs by providing contextual examples to facilitate more accurate responses to queries. Typically, ICL involves two primary steps: (1) retrieving demonstration examples from a database that are relevant to the query, and (2)

incorporating these samples as contextual input preceding the query. Numerous approaches have been proposed to enhance the retrieval process and consequently improve the quality of responses generated by LLMs (Ye et al., 2023; Li et al., 2023).

However, nearly all research on ICL assumes that the underlying database of descriptions is entirely factual. Only a limited number of studies have addressed scenarios in which the database contains incorrect information, often referred to as noisy attributes. For instance, in Gao et al. (2024), the authors proposed a local perplexity ranking method to identify noisy information and replace demonstrations deemed to be noisy. Similarly, the authors in Kang et al. (2024) introduced an algorithm called *rectification*, which refines the labels by fine-tuning pre-trained LLMs, such as GPT-2 (Radford et al., 2019).

While these approaches have explored this meaningful problem setting and shown robust results under demonstrations with noisy annotations, they still present certain limitations. First, the previous methods presuppose that clean demonstrations make up a substantial portion of the training set, leading to methodological failures when the noise ratio is high. Second, the reliance on perplexity, or similar metrics, is predicated on the assumption that noisy demonstrations consistently manifest higher perplexity than clean ones. Lastly, existing studies have not attempted to articulate the influence of the LLM’s prior knowledge on their ability to detect the matching relationship between query and annotation of demonstration samples.

In this paper, we begin by revisiting the naive probability-based metric, which depicts LLM’s perception on the matching relationship between queries and annotations in the noisy ICL generation task. We further investigate the potential influence of LLM’s prior knowledge on the values of this metric, which we categorize as intrinsic bias and extrinsic bias. Based on our findings, we mathe-

\*Corresponding author.

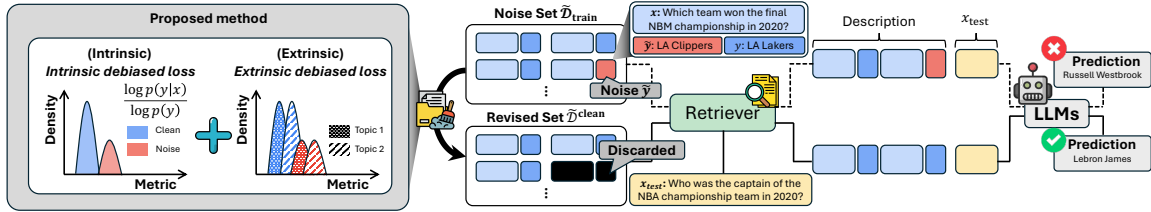


Figure 1: Overview of noisy ICL and the proposed method: When noisy information is present in an ICL dataset, it can lead to performance degradation (Top case). However, if the noisy information is sufficiently removed, performance can be improved (Bottom case). In this paper, we go beyond the conventional perplexity score-based approach and propose a Sample Cleanliness Score, which is based on intrinsic/extrinsic debiased loss.

matically formulate both types of bias, providing a quantified assessment of the impact of LLM’s prior knowledge on their ability to discern the matching relationship between query and annotation of each demonstration sample.

Building on these insights, we introduce an explicit dual-debiasing method that leverages neighbor-based feasible estimation to mitigate the effects of both intrinsic and extrinsic biases. We then present the *Sample Cleanliness Score*, a metric developed to detect noisy demonstrations, and our pipeline incorporating the score as a solution to the noisy challenge in ICL.

We describe our key contributions as follows:

- We reveal the potential influence of LLM’s prior knowledge on their perception of the query-annotation matching relationship, specifically identifying *intrinsic bias* and *extrinsic bias*, and we provide their mathematical formulation.
- We propose practical approximations for intrinsic and extrinsic biases, utilizing a neighbor-based method for the latter, to enable effective assessments of these biases.
- We propose a novel dual-debiasing approach to mitigate the impact of LLM’s prior knowledge on their perception of the query-annotation matching relationship. This method leads to developing the *sample cleanliness score*, a new metric designed to detect noisy demonstrations.
- We design a metric-based pipeline for addressing the challenges of noisy ICL, which is sufficiently robust even under extreme noise cases.
- We evaluate the efficacy of our pipeline across diverse benchmark datasets for ICL text generation tasks under various noisy settings. Our results show superior performance compared with

several baselines, with outcomes in many cases comparable to those achieved in clean settings.

## 2 Preliminaries

We consider ICL in text generation tasks. Given the training set  $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$ , where  $x_i$  is the demonstration query text and  $y_i$  is the tokenized corresponding annotation with length  $T_i = |y_i|$ , ICL aims to utilize a LLM to generate sequence output for test queries in the test set  $\mathcal{D}_{\text{test}} = \{x_j^{\text{test}}\}_{j=1}^M$ .

A typical ICL process contains retrieval and inference steps. Given a test query  $x^{\text{test}} \in \mathcal{D}_{\text{test}}$ , the retriever retrieves  $k$  demonstration examples from  $\mathcal{D}_{\text{train}}$ , say,  $\mathcal{D}_{\text{ex}} = \{(x_i, y_i)\}_{i \in \mathcal{S}}$ , where  $\mathcal{S}$  is the index set of retrieved samples with  $|\mathcal{S}| = k$ . Then the prompt  $\mathcal{P}$  will be constructed using the retrieved examples  $\mathcal{D}_{\text{ex}}$  and the given test query  $x^{\text{test}}$  based on the prompt template  $\mathcal{T}$ . By feeding the constructed prompt  $\mathcal{P}$  into LLM for inference, we can obtain generated results via:

$$y_t \sim P_{\text{LLM}}(Y_t | \mathcal{P}, y_{<t}),$$

where  $\sim$  denotes the decoding strategies. ICL performance relies on the quality of retrieved demonstration examples (Li et al., 2023; Ye et al., 2023).

However, the training set in real-world settings can easily include noised annotations  $\tilde{\mathcal{D}}_{\text{train}} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$ , due to unreliable data sources or limited annotation expertise. Thus, the retrieved demonstration examples  $\tilde{\mathcal{D}}_{\text{ex}} = \{(x_i, \tilde{y}_i)\}_{i \in \mathcal{S}}$  can introduce misleading information in the prompt, which thus leads to degraded performance in ICL (Yoo et al., 2022; Gao et al., 2024).

Though previous work by Gao et al. (2024) has identified the issue of noise in ICL for text generation tasks and suggested potential solutions, it presupposes that clean samples predominate in the training set. This leads to the failure of the pro-

posed method when the noise ratio is high. Additionally, though the authors acknowledged the limitations of naive perplexity in distinguishing between noisy and clean demonstrations due to inherent perturbation, their method only mitigates this effect implicitly. It does not thoroughly examine the underlying mechanism of the query-annotation relationship for demonstration samples. This gap in understanding motivates our further investigation into ICL generation tasks with noisy annotations to develop a more generalizable solution. Furthermore, it requires a deeper comprehension of the influence of LLM’s parametric knowledge on its detection capability of noisy matching relations in query-annotation pairs.

### 3 Method

In this section, we will first introduce our metric design motivation. Then, we deliver the explicit dual-debiasing method to compute the *Sample Cleanliness Score* for each demonstration sample from the given training set, including the intrinsic-debiasing step and the neighbor-based extrinsic-debiasing step. Finally, we introduce the complete pipeline for noisy ICL utilizing proposed *Sample Cleanliness Score*.

#### 3.1 Motivation on Probability

Intuitively, a well-pretrained LLM is more likely to assign higher probability to the correct annotation (or, the in-distribution annotation) than the noised one (or, out-of-distribution one), when conditioned on the same query  $\mathbf{x}$  (Arora et al., 2021; Alon and Kamfonas, 2023; Gao et al., 2024), that is,

$$P(\mathbf{y}^*|\mathbf{x}) > P(\tilde{\mathbf{y}}|\mathbf{x}),$$

where  $\mathbf{y}^*$  with length  $T^*$  is the correct annotation for  $\mathbf{x}$ , and  $\tilde{\mathbf{y}} \neq \mathbf{y}^*$  with length  $\tilde{T}$  is the observed noised annotation. Due to the impact of varied token sequence lengths, we consider the following token-wise version of the conditional probabilities:

$$P(\mathbf{y}^*|\mathbf{x})^{1/T^*} > P(\tilde{\mathbf{y}}|\mathbf{x})^{1/\tilde{T}}. \quad (1)$$

Now, applying the logarithmic transformation to both sides of Equation 1, we have:

$$-\frac{1}{T^*} \log P(\mathbf{y}^*|\mathbf{x}) < -\frac{1}{\tilde{T}} \log P(\tilde{\mathbf{y}}|\mathbf{x}). \quad (2)$$

Given  $P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^T P(y_t|\mathbf{x}, y_{<t})$ , we define the following based on per-token conditioned prob-

abilities for sequence  $\mathbf{y}$  given prefix sequence  $\mathbf{x}$ :

$$\mathcal{L}(\mathbf{y}|\mathbf{x}) = -\frac{1}{T} \sum_{t=1}^T \log P(y_t|\mathbf{x}, y_{<t}), \quad (3)$$

which is the per-token version of Negative Log-Likelihood (NLL) loss for  $\mathbf{y}$  given prefix  $\mathbf{x}$  and can be easily computed using next-token probabilities output from LLM.

Equation 2 can thus be rewritten as  $\mathcal{L}(\mathbf{y}^*|\mathbf{x}) < \mathcal{L}(\tilde{\mathbf{y}}|\mathbf{x})$  when  $\tilde{\mathbf{y}} \neq \mathbf{y}^*$ . This suggests that for a fixed query token sequence  $\mathbf{x}$ , the noised annotation is expected to exhibit a higher per-token NLL loss value than its clean counterpart.

However, directly using  $\mathcal{L}(\mathbf{y}|\mathbf{x})$  does a poor job on differentiating noisy and clean demonstrations. We provide this analysis in subsection B.2. In short, the distribution of clean demonstrations’ per-token NLL loss values overlaps heavily with that of noisy ones, which makes it almost impossible to determine whether a demonstration is clean or not given its  $\mathcal{L}(\mathbf{y}|\mathbf{x})$  value alone.

This can be attributed to bias factors like the demonstration sample itself, LLM’s prior knowledge (Fei et al., 2023; Zhao et al., 2021), and even LLM architectures (O’Brien and Lewis, 2023; Li et al., 2022). The previous paper (Gao et al., 2024) also noticed a similar phenomenon on sample-wise perplexity. Unlike their method, which tries to disentangle the perplexity impact of LLMs implicitly, we propose the dual-debiasing method to remove the bias derived from the demonstration itself, which we call *intrinsic debiasing*, as well as the bias derived from various expertise levels of LLM’s parametric knowledge on different domains, which we call *extrinsic debiasing*. Our ultimate objective is to formulate a metric for each demonstration sample  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i) \sim \tilde{\mathcal{D}}_{\text{train}}$  that satisfies two key properties: 1) it accesses the *matching relationship between the annotation and the query* of each demonstration; 2) it is *comparable across different demonstration samples*, regardless of variations in both query  $\mathbf{x}_i$  and annotation  $\tilde{\mathbf{y}}_i$ . We intend for this metric to facilitate the determination of whether a demonstration is clean or noisy based on its metric value.

#### 3.2 Intrinsic Debiasing

When we only consider  $\mathcal{L}(\tilde{\mathbf{y}}|\mathbf{x})$  to evaluate the matching relationship between  $\mathbf{x}$  and observed  $\tilde{\mathbf{y}}$ , the pre-trained LLM may be very familiar with  $\tilde{\mathbf{y}}$  itself, given the frequent occurrence of  $\tilde{\mathbf{y}}$  in the pre-training dataset. This will lead to  $\mathcal{L}(\tilde{\mathbf{y}}|\mathbf{x}) <$

$\mathcal{L}(\mathbf{y}^*|\mathbf{x})$  even when the observed  $\tilde{\mathbf{y}}$  does not match with  $\mathbf{x}$  as noised annotation, given LLM assigns high probability on  $\tilde{\mathbf{y}}$  than  $\mathbf{y}^*$  without any prefix token sequence. In other words, the naive  $\mathcal{L}(\mathbf{y}|\mathbf{x})$  is biased by LLM’s prior knowledge on annotation  $\mathbf{y}$ . Since this bias is derived from the annotation part of the demonstration sample itself, we name it *intrinsic bias*.

Motivated by this, we propose the *intrinsic debiasing* step to remove LLM’s prior knowledge bias on  $\mathbf{y}$ , i.e.,  $P(\mathbf{y})$ . We define the per-token loss function for sequence  $\mathbf{y}$  without any prefix as:

$$\mathcal{L}(\mathbf{y}) = -\frac{1}{T} \sum_{t=1}^T \log P(y_t|y_{<t}),$$

which serves as an effective alternative for representing  $P(\mathbf{y})$ . Then we defined the intrinsic-debiased per-token loss function as:

$$\mathcal{L}_{\text{de-int}}(\mathbf{y}|\mathbf{x}) = \frac{\mathcal{L}(\mathbf{y}|\mathbf{x})}{\mathcal{L}(\mathbf{y})}. \quad (4)$$

Given a fixed demonstration query  $\mathbf{x}$ , a clean annotation is expected to exhibit a lower value of  $\mathcal{L}_{\text{de-int}}(\mathbf{y}|\mathbf{x})$  than a noised one. Specifically, if  $\tilde{\mathbf{y}} \neq \mathbf{y}^*$ , then

$$\mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}|\mathbf{x}) > \mathcal{L}_{\text{de-int}}(\mathbf{y}^*|\mathbf{x}). \quad (5)$$

Furthermore, in cases where the ground-truth annotation is unavailable and only two observed annotations,  $\tilde{\mathbf{y}}_1$  and  $\tilde{\mathbf{y}}_2$ , are provided for a given  $\mathbf{x}$ , it can be inferred that  $\tilde{\mathbf{y}}_1$  is more mismatching with  $\mathbf{x}$  than  $\tilde{\mathbf{y}}_2$  if  $\mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}_1|\mathbf{x}) > \mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}_2|\mathbf{x})$ .

### 3.3 Neighbor-based Extrinsic Debiasing

Using  $\mathcal{L}_{\text{de-int}}$ , we can assess the relative query-annotation mismatch between two demonstrations that share the same query  $\mathbf{x}$ . However, within the dataset  $\tilde{\mathcal{D}}_{\text{train}}$ , each query is associated with only one annotation - either clean or noisy, but never both simultaneously. This raises an important question: Are the intrinsic-debiased per-token loss values comparable when both queries and annotations differ? Specifically, given pairs  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$  and  $(\mathbf{x}_j, \tilde{\mathbf{y}}_j)$  with  $\mathbf{x}_i \neq \mathbf{x}_j$  and  $\tilde{\mathbf{y}}_i \neq \tilde{\mathbf{y}}_j$ , can we conclude that  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$  is more likely to be noisy than  $(\mathbf{x}_j, \tilde{\mathbf{y}}_j)$  if  $\mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}_i|\mathbf{x}_i) > \mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}_j|\mathbf{x}_j)$ ?

LLMs exhibit varying levels of expertise across different knowledge domains. For instance, GPT-4 (Achiam et al., 2023) exhibits diverse capabilities in translation tasks, as demonstrated in recent benchmark studies (Yan et al., 2024). Moreover,

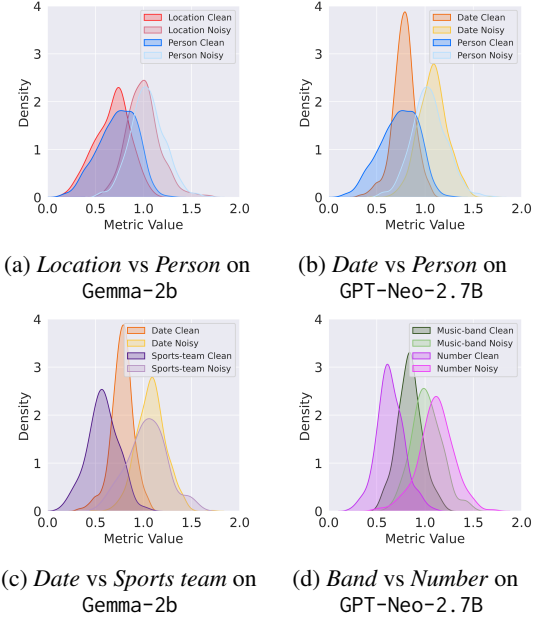


Figure 2: Distribution of  $\mathcal{L}_{\text{de-int}}$  for both clean and noisy versions of demonstrations for different topics. The  $\mathcal{L}_{\text{de-int}}$  values are calculated using Gemma-2b and GPT-Neo-2.7B.

LLMs show different degrees of familiarity with queries from distinct knowledge domains, which affects the value distributions of  $\mathcal{L}_{\text{de-int}}$  for samples from these domains. Consequently, the  $\mathcal{L}_{\text{de-int}}$  values of different domains are not directly comparable, making it challenging to compare samples such as  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$  and  $(\mathbf{x}_j, \tilde{\mathbf{y}}_j)$  when they originate from different knowledge domains.

Motivated by this, we categorize the training set of NQ (Kwiatkowski et al., 2019) dataset into different sample topics using GPT-4, and visualize the  $\mathcal{L}_{\text{de-int}}$  value distributions for both noisy and clean demonstration samples across different topics, including *Location*, *Person*, *Date*, *Music-Band*, *Number* and *Sports-Team*. We use both Gemma-2b (Team et al., 2024) and GPT-Neo-2.7B (Black et al., 2021) to calculate the  $\mathcal{L}_{\text{de-int}}$  values for demonstrations.

As shown in Figure 2a and Figure 2b, within each topic, the value distribution of noisy demonstrations is significantly separated from that of clean ones, and clean demonstrations maintain similar value distributions across different topics. Specifically, Figure 2a reveals that clean/noisy demonstrations from *Location* share similar distributions with clean/noisy demonstrations from *Person* under Gemma-2b. Similarly, Figure 2b shows this pattern for *Date vs Person* under GPT-Neo-2.7B. In these cases, we can directly

compare  $\mathcal{L}_{\text{de-int}}$  values between demonstrations from different topics to determine their relative query-annotation mismatching levels.

However, different patterns emerge when examining demonstrations from *Date vs Sports-Team* under Gemma-2b, and *Music-Band vs Number* under GPT-Neo-2.7B. As shown in Figure 2c, under Gemma-2b, while clean and noisy demonstrations are well-separated within each topic, the mean value of *Date* clean demonstration distribution is significantly higher than that of *Sports-Team* clean demonstrations, leading to more distribution overlap with *Sports-Team* noisy demonstrations. This distribution pattern can lead to false noise detection for *Date* clean demonstrations, as they may be incorrectly classified as noisy when their  $\mathcal{L}_{\text{de-int}}$  values are compared with *Sports-Team* ones. We observe a similar issue between *Music-Band vs Number* under Gemma-2b as shown in Figure 2d.

To conclude, LLMs’ bias across different knowledge domains makes  $\mathcal{L}_{\text{de-int}}$  values incomparable for demonstration samples from different domains. We term this domain-related bias as *extrinsic bias* since it originates from the demonstration sample’s domain rather than the sample itself.

Based on these observations, we propose the *extrinsic debiasing* step to eliminate bias associated with different knowledge domains. This step aims to facilitate comparability of the metric values across demonstrations, even when they originate from distinct domains. Given data point <sup>1</sup>  $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ , we define an associated domain based on a metric space  $(\mathcal{X} \times \mathcal{Y}, d)$  with a distance function  $d$ :

$$\mathcal{N}((\mathbf{x}, \mathbf{y})) = \{(\mathbf{x}', \mathbf{y}') \in \mathcal{X} \times \mathcal{Y} : d((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) < \eta\}, \quad (6)$$

where  $d$  is a suitable distance metric, and  $\eta$  is a positive real number indicating the radius of the associated domain. For ease of notation and where the context is clear, we will subsequently denote  $\mathcal{N}((\mathbf{x}, \mathbf{y}))$  simply as  $\mathcal{N}$ . Assume the density function  $\rho$  is uniform over given defined domain  $\mathcal{N}((\mathbf{x}, \mathbf{y}))$  with volume  $V$ :

$$\rho(\mathbf{x}', \mathbf{y}') = \frac{1}{V}, \quad \text{if } (\mathbf{x}', \mathbf{y}') \in \mathcal{N}((\mathbf{x}, \mathbf{y})),$$

where  $V$  is the measure of the associate domain defined by  $V = \int_{(\mathbf{x}', \mathbf{y}') \in \mathcal{N}} dV$  and  $dV$  is the differential volume element in the continuous space

<sup>1</sup>The definition of  $(\mathbf{x}, \mathbf{y})$  here is not on the discrete demonstration sample space, but in the continuous space.

$\mathcal{X} \times \mathcal{Y}$ . To this end, we can estimate the *extrinsic bias*, denoted as  $\Phi((\mathbf{x}, \mathbf{y}))$ , over the domain of  $(\mathbf{x}, \mathbf{y})$  using the empirical estimation of  $\mathcal{L}_{\text{de-int}}$ :

$$\Phi((\mathbf{x}, \mathbf{y})) = \frac{1}{V} \int_{(\mathbf{x}', \mathbf{y}') \in \mathcal{N}} \mathcal{L}_{\text{de-int}}(\mathbf{y}'|\mathbf{x}') dV. \quad (7)$$

Then we can perform the extrinsic debiasing via  $\mathcal{L}_{\text{de-int}}(\mathbf{y}|\mathbf{x})/\Phi((\mathbf{x}, \mathbf{y}))$ , which equivalently, we introduce the final metric used for each demonstration sample, *i.e.*, *Sample Cleanliness Score*, as

$$\mathcal{I}(\mathbf{x}, \mathbf{y}) = \frac{\Phi((\mathbf{x}, \mathbf{y}))}{\mathcal{L}_{\text{de-int}}(\mathbf{y}|\mathbf{x})}. \quad (8)$$

Given two demonstration samples  $(\mathbf{x}_1, \tilde{\mathbf{y}}_1)$  and  $(\mathbf{x}_2, \tilde{\mathbf{y}}_2)$  from  $\tilde{\mathcal{D}}_{\text{train}}$ , if  $\tilde{\mathbf{y}}_1$  is clean while  $\tilde{\mathbf{y}}_2$  is noised, the following relation holds

$$\mathcal{I}(\mathbf{x}_1, \tilde{\mathbf{y}}_1) > \mathcal{I}(\mathbf{x}_2, \tilde{\mathbf{y}}_2),$$

even when two samples are from different domains.

Note that Equation 7 is defined on the continuous space, which is intractable to estimate. Thus, we propose to use the finite discrete neighboring demonstration samples to provide a tractable estimation. We provide the following construction of neighboring demonstration samples to serve as alternation for the domain  $\mathcal{N}$ :

$$\mathcal{N}_{\text{DISC}}((\mathbf{x}, \mathbf{y})) = \{(\mathbf{x}, \mathbf{y}'_z)\}_{z=1}^{N_{\text{neighbor}}}, \quad (9)$$

where  $\mathbf{y}'_z$  can be tokenized sequence sampled from a large corpus  $\mathcal{C}$ , and  $N_{\text{neighbor}}$  is the number of neighbors. We denote  $\mathcal{N}_{\text{DISC}}((\mathbf{x}, \mathbf{y}))$  simply as  $\mathcal{N}_{\text{DISC}}$  for ease of notation. We can define the distance function  $d(\cdot, \cdot)$  based on Edit Distance  $d_{\text{edit}}$ , which can help us to bound the radius of  $\mathcal{N}_{\text{DISC}}$ :

$$\begin{aligned} d((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) &= d((\mathbf{x}, \mathbf{y}), (\mathbf{x}, \mathbf{y}')) \\ &= d_{\text{edit}}(\mathbf{y}, \mathbf{y}') \\ &\leq \max(T, T_{\text{max}}) = \eta \\ &\text{for } \forall(\mathbf{x}', \mathbf{y}') \in \mathcal{N}_{\text{DISC}}((\mathbf{x}, \mathbf{y})), \end{aligned}$$

where  $T_{\text{max}}$  is the maximum length of sequences in  $\mathcal{C}$ . And Equation 7 can be replaced by:

$$\Phi((\mathbf{x}, \mathbf{y})) = \frac{\sum_{(\mathbf{x}', \mathbf{y}') \sim \mathcal{N}_{\text{DISC}}} \mathcal{L}_{\text{de-int}}(\mathbf{y}'|\mathbf{x}')}{N_{\text{neighbor}}}. \quad (10)$$

Consequently, *Sample Cleanliness Score*  $\mathcal{I}(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$  is easy to calculate using neighbor-based extrinsic debiasing step for each  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i) \in \tilde{\mathcal{D}}_{\text{train}}$ . We simply use  $\mathcal{I}_i$  for  $\mathcal{I}(\mathbf{x}_i, \tilde{\mathbf{y}}_i)$  for short if context is clear.

We depict the distribution of the proposed  $\mathcal{I}$  for both clean and noisy samples in Figure 3. The results show that we can now effectively differentiate noisy samples from clean ones based on the metric values.

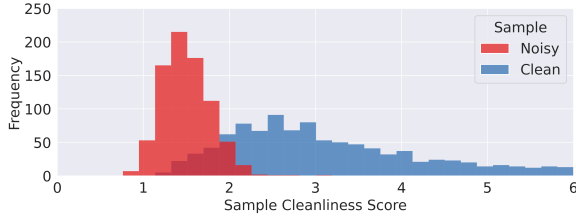


Figure 3: Sample Cleanliness Score distributions on noisy NQ training set. A larger value indicates that it is more likely to be a clean sample. It shows the pattern of distinguishability between noisy and clean samples.

### 3.4 Complete Noisy ICL Pipeline

We now introduce our complete noisy ICL pipeline based on the proposed *Sample Cleanliness Score*. The pipeline contains three steps: 1) noisy demonstration detection based on the proposed metric; 2) noisy demonstration cleanse; 3) regular ICL process. We provide detailed pseudocode for our noisy ICL pipeline in algorithm 1 of Appendix C.

Our initial step involves the Gaussian Mixture Model (GMM) to detect noisy demonstrations following the classic noisy label learning routine (Li et al.; Huang et al., 2023; Karim et al., 2022). We fit a 2-component GMM on  $\{\mathcal{I}_i\}_{i=1}^N$ , calculated for samples of the whole demonstration set, using the Expectation-Maximization algorithm. Then calculate the noisy probability of each demonstration sample  $q_i$  as the posterior probability  $q(g_{\text{noisy}} | \mathcal{I}_i)$ , that indicates the possibility of  $\mathcal{I}_i$  belongs to the Gaussian component  $g_{\text{noisy}}$  representing the noisy demonstration samples, with a lower mean value of *Sample Cleanliness Score*. Using threshold  $\gamma$  on  $q_i$  on all demonstration samples to separate the whole training set  $\tilde{\mathcal{D}}_{\text{train}}$  into clean subset  $\tilde{\mathcal{D}}^{\text{clean}}$  and noisy subset  $\tilde{\mathcal{D}}^{\text{noisy}}$ .

The second step is to handle the detected clean subset  $\tilde{\mathcal{D}}^{\text{clean}}$  and noisy subset  $\tilde{\mathcal{D}}^{\text{noisy}}$  obtained from previous step. One can follow (Gao et al., 2024) to replace each detected noisy demonstration with the nearest clean ones from  $\tilde{\mathcal{D}}^{\text{clean}}$ . Yet the powerful denoising capability of the proposed metric suggests an alternative approach that eliminates all identified noisy samples, which achieves efficiency by reducing the size of the final training set for sample retrieval. Our experimental results show

that the simple strategy achieves surprising ICL performance, even comparable to a clean setting.

The concluding phase of our pipeline entails the standard ICL process, which encompasses retrieval and inference, as outlined in section 2. However, in this phase, the original training set for retrieval is substituted with the identified clean subset  $\tilde{\mathcal{D}}^{\text{clean}}$ .

## 4 Experiment

### 4.1 Experiment Setting

**Datasets:** We evaluate the performance of the proposed algorithm using four different datasets: (1) **Natural Questions (NQ)**: a large-scale dataset containing real user queries/ from Google Search, paired with human-annotated answers. (2) **Web Questions (WebQ)**: a dataset consisting of questions posed to Google Search, with answers derived from Freebase. (3) **SciQ**: a science-focused dataset with multiple-choice questions covering physics, chemistry, and biology. (4) **SQuAD**: a reading comprehension dataset containing questions and answers based on Wikipedia passages.

**Language Models:** We use Llama-2-7B (Touvron et al., 2023) as the default ICL inference LLM. For the calculation of metrics, we use Llama-2-7B as the default model, and use GPT-Neo-1.3B (Black et al., 2021), Gemma-2b (Team et al., 2024) and Mistral-7B-v0.1 (Jiang et al., 2023) for analysis experiments.

**Implementation details:** We implement our noisy ICL pipeline and the baselines based on OpenICL (Wu et al., 2023). We use Random, TopK, and DPP (Ye et al., 2023) retrievers for the retrieval process. We follow Gao et al. (2024) for the noise generation process for both relevant/irrelevant noise. And the default hyperparameters for noise detection setting are  $\gamma = 0.5$ ,  $N_{\text{neighbor}} = 50$ . In the construction of  $\mathcal{N}_{\text{DISC}}$ , we deploy two implementations by utilizing two distinct corpora  $\mathcal{C}$ , for the sampling of  $\mathbf{y}'$ . The first is termed the in-distribution corpus  $\mathcal{C}_{\text{in}}$  which is compiled from all annotations within the observed  $\tilde{\mathcal{D}}_{\text{train}}$ ; the second is designated as the out-distribution corpus,  $\mathcal{C}_{\text{out}}$ , and is comprised of annotations from external datasets. To enhance the estimation of extrinsic bias, we constraint  $T_{\text{max}}$  to equal the maximum annotation length in  $\tilde{\mathcal{D}}_{\text{train}}$ , thereby bounding the radius  $\eta$  of demonstration sample’s associated domain to  $T_{\text{max}}$ . We provide

a detailed description of the generation process of neighbors in Appendix D. For the main results, we report the optimal performance achieved by either  $\mathcal{C}_{in}$  or  $\mathcal{C}_{out}$ . However,  $\mathcal{C}_{out}$  is exclusively used as the default implementation for analytical experiments. Furthermore, to save the computation costs, we employ a fixed set of  $\mathbf{y}'$  for all demonstration samples within the same dataset, thereby decreasing the computational expenditure required for estimating the extrinsic bias  $\Phi$ .

**Baselines:** We evaluate the performance of our framework alongside three baselines: (1) Naive ICL: the conventional ICL pipeline without employing any specialized cleansing method, meaning that noisy information may be included in the description. (2) Random delete: a method that removes a randomly selected subset of samples corresponding to the noise ratio. (3) LPR (Gao et al., 2024): a method that leverages perplexity to cleanse the description using a local perplexity ranking score. (4) Ours: the proposed cleansing method utilizing the dual-debiasing approach.

## 4.2 Main results

As shown in Table 1, we see a substantial improvement in the performance of the proposed algorithm across all degrees of noise and all retriever types. In particular, when the noise level is high, *i.e.*, 0.8, our algorithm outperforms the naive ICL approach (*i.e.*, without any robust ICL method) by the largest margin. For example, on the SCIQ dataset with 0.8 irrelevant noise, using the TopK retriever improves performance by 37.94. Moreover, under the same setting for relevant noise, we observe an increase of 26.49. These results indicate that the proposed algorithm can operate robustly under noisy ICL.

## 4.3 Analysis

**Different LLMs for noise detection.** We utilize three other distinct LLMs to calculate the  $\{\mathcal{I}_i\}_{i=1}^N$  for noise detection and assess the final ICL performance. We employ GPT-Neo-1.3B (Black et al., 2021) as a representative of smaller and weaker LLMs, Gemma-2b (Team et al., 2024) as a smaller yet potent LLM, and Mistral-7B-v0.1 (Jiang et al., 2023) to represent LLMs of comparable size with strong capabilities, compared with our default metric model Llama-2-7B (Touvron et al., 2023). As illustrated in Table 2, even a relatively small and less capable LLM such as GPT-Neo-1.3B exhibits only a negligible performance decline in both rele-

vant and irrelevant noise settings, even with a high noise ratio of 0.6. This shows the robustness of our method, even when applied using smaller, less capable LLMs. Additionally, the stability of our method with smaller LLMs suggests that it can be generalized to scenarios with severe computational constraints, employing smaller LLMs to compute  $\{\mathcal{I}_i\}_{i=1}^N$ . We also compare the AUC gain for noise detection by applying our dual-debiasing method on different LLMs as shown in Table 7 of subsection B.4, which further reveals that our method is particularly valuable for resource-constrained scenarios.

Dataset	Retriever	Metric Model	Irrelevant Noise		Relevant Noise	
			0.4	0.6	0.4	0.6
NQ	Random	GPT-Neo-1.3B	13.87	13.10	14.13	12.87
		Gemma-2b	13.93	13.84	13.80	13.00
		Mistral-7B-v0.1	13.80	13.37	13.87	13.57
		Llama-2-7B	13.93	13.67	13.33	13.73
	TopK	GPT-Neo-1.3B	15.47	15.00	14.07	13.80
		Gemma-2b	16.20	16.20	16.00	15.77
		Mistral-7B-v0.1	16.00	16.54	16.47	16.14
		Llama-2-7B	16.33	15.70	15.87	15.64
	DPP	GPT-Neo-1.3B	16.53	15.54	14.07	13.84
		Gemma-2b	16.33	16.00	16.13	14.73
		Mistral-7B-v0.1	16.20	15.80	16.47	15.40
		Llama-2-7B	16.40	16.34	16.47	15.74
SciQ	Random	GPT-Neo-1.3B	75.69	75.29	75.09	74.71
		Gemma-2b	76.04	75.69	76.04	75.69
		Mistral-7B-v0.1	76.18	76.01	75.58	75.98
		Llama-2-7B	75.84	75.43	75.98	76.41
	TopK	GPT-Neo-1.3B	73.45	75.00	74.43	74.00
		Gemma-2b	73.42	74.57	74.17	74.34
		Mistral-7B-v0.1	73.60	75.37	74.20	75.43
		Llama-2-7B	73.57	75.00	73.82	75.18
	DPP	GPT-Neo-1.3B	74.25	74.20	75.40	74.20
		Gemma-2b	73.97	74.66	74.37	73.22
		Mistral-7B-v0.1	74.37	75.03	74.31	74.83
		Llama-2-7B	74.31	74.83	74.43	74.37

Table 2: ICL performance of using different LLM as the metric model. The default inference LLM is Llama-2-7B.

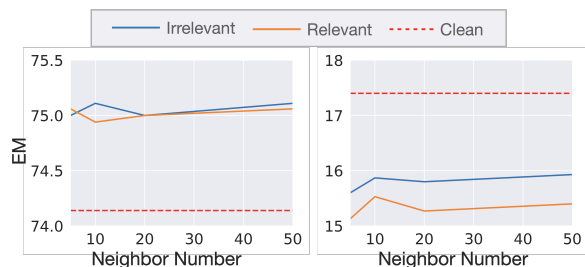


Figure 4: Sensitivity on neighbor number  $N_{neighbor}$  under noise ratio 0.6 and TopK retriever. Left: SCIQ; right: NQ. The ‘irrelevant’/‘relevant’ indicates the performance of our method under noise, and ‘clean’ indicates the performance of naive baseline under clean.

**Computation cost.** We analyze the proposed algorithm based on two key factors. First, we compare its performance when using different metric

Dataset	Retriever	Method	Clean	Irrelevant Noise				Relevant Noise				Dataset	Retriever	Method	Clean	Irrelevant Noise				Relevant Noise			
				0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8					0.2	0.4	0.6	0.8				
NQ	Random	Naive ICL	<b>14.00</b>	12.07	9.20	6.87	3.67	12.13	11.87	10.60	5.33	SciQ	Random	Naive ICL	74.83	68.74	56.84	39.83	20.29	73.62	71.44	65.29	56.38
		Random	-	11.73	9.53	6.53	4.53	13.13	12.40	10.40	8.67			-	68.79	58.79	40.57	22.64	73.10	70.75	66.15	57.13	
		LPR	12.80	11.67	10.47	8.47	5.33	11.53	10.33	9.73	7.33			LPR	69.43	64.60	54.77	40.06	20.92	66.38	61.90	56.95	44.60
		Ours	13.40	<b>13.47</b>	<b>13.93</b>	<b>14.00</b>	<b>13.60</b>	<b>13.93</b>	<b>13.33</b>	<b>13.73</b>	<b>12.33</b>			Ours	<b>75.98</b>	<b>75.23</b>	<b>75.98</b>	<b>74.77</b>	<b>74.83</b>	<b>75.46</b>	<b>75.17</b>	<b>76.03</b>	<b>75.63</b>
	TopK	Naive ICL	<b>17.40</b>	15.07	10.80	9.33	5.33	15.93	13.53	12.33	7.33	TopK	Naive ICL	74.14	67.18	51.78	36.95	18.56	71.44	64.20	58.45	48.22	
		Random	-	15.47	11.20	8.67	5.27	16.20	13.67	11.53	8.27		Random	-	68.05	54.02	38.16	20.75	72.01	67.07	60.46	52.24	
		LPR	12.73	12.84	11.87	10.33	6.67	12.67	11.13	9.47	7.40		LPR	69.20	64.14	57.01	41.21	22.82	66.38	62.64	53.28	40.11	
		Ours	16.33	<b>16.40</b>	<b>16.33</b>	<b>15.93</b>	<b>15.40</b>	<b>16.00</b>	<b>15.60</b>	<b>15.87</b>	<b>13.80</b>		Ours	<b>74.66</b>	<b>74.08</b>	<b>73.62</b>	<b>74.89</b>	<b>75.34</b>	<b>74.20</b>	<b>73.85</b>	<b>75.06</b>	<b>74.71</b>	
	DPP	Naive ICL	<b>18.33</b>	16.07	11.00	8.80	5.47	15.73	14.07	11.27	9.87	DPP	Naive ICL	74.13	66.84	52.70	36.78	20.17	71.67	66.38	61.72	51.38	
		Random	-	14.87	11.93	7.80	5.00	15.87	13.73	10.93	8.87		Random	-	67.70	56.09	38.85	21.72	72.47	68.39	61.61	55.34	
		LPR	13.07	13.40	12.47	10.47	7.20	12.67	11.13	9.47	7.40		LPR	67.64	65.29	56.78	43.68	24.83	65.75	62.76	54.48	44.94	
		Ours	16.93	<b>16.13</b>	<b>16.40</b>	<b>16.60</b>	<b>14.80</b>	<b>15.93</b>	<b>16.47</b>	<b>16.20</b>	<b>14.67</b>		Ours	<b>74.94</b>	<b>74.25</b>	<b>74.31</b>	<b>74.71</b>	<b>74.43</b>	<b>73.97</b>	<b>74.43</b>	<b>74.37</b>	<b>74.83</b>	
Random	Naive ICL	22.64	17.97	11.84	7.72	3.65	19.37	16.52	12.45	9.40	Random	Naive ICL	34.70	34.97	32.87	26.53	19.67	34.27	32.77	30.93	29.20		
	Random	-	18.25	12.34	7.34	3.96	20.50	15.94	13.30	8.93		Random	-	34.93	31.60	26.67	18.73	33.90	33.87	30.67	28.43		
	LPR	22.04	17.70	13.82	9.12	3.82	17.64	14.15	10.58	7.53		LPR	28.27	28.03	27.80	24.67	19.73	27.30	27.00	26.13	24.40		
	Ours	<b>23.41</b>	<b>23.19</b>	<b>23.06</b>	<b>23.14</b>	<b>23.22</b>	<b>23.17</b>	<b>22.20</b>	<b>21.13</b>	<b>19.62</b>		Ours	<b>35.17</b>	<b>34.77</b>	<b>35.53</b>	<b>35.57</b>	<b>35.20</b>	<b>34.83</b>	<b>35.73</b>	<b>35.67</b>	<b>35.20</b>		
WebQ	TopK	Naive ICL	<b>44.11</b>	34.98	24.92	15.77	7.09	36.55	29.60	21.19	14.48	SQuAD	Naive ICL	34.47	33.40	30.53	24.63	17.63	33.47	31.47	29.37	26.03	
		Random	-	32.89	20.47	11.95	5.94	34.73	26.38	18.71	12.39		Random	-	32.97	30.37	26.10	17.43	33.53	31.97	30.87	26.63	
		LPR	26.49	22.81	19.48	14.37	6.35	21.19	18.33	14.10	9.59		LPR	34.87	26.13	25.27	21.93	17.67	25.50	25.23	24.00	21.93	
		Ours	36.33	<b>37.76</b>	<b>35.72</b>	<b>34.13</b>	<b>29.29</b>	<b>36.85</b>	<b>33.17</b>	<b>30.92</b>	<b>26.30</b>		Ours	<b>35.33</b>	<b>35.57</b>	<b>35.73</b>	<b>36.80</b>	<b>36.37</b>	<b>36.07</b>	<b>35.90</b>	<b>35.97</b>	<b>35.83</b>	
	DPP	Naive ICL	<b>45.12</b>	36.60	25.94	16.21	7.53	37.54	30.01	21.60	14.81	DPP	Naive ICL	36.47	35.23	31.83	25.93	17.73	35.87	33.87	30.93	27.37	
		Random	-	34.76	21.74	11.62	5.99	36.33	26.94	17.61	11.54		Random	-	34.90	31.73	25.73	17.77	35.40	33.97	30.63	27.27	
		LPR	26.68	22.23	18.88	12.97	6.10	21.52	17.78	14.21	8.93		LPR	26.03	26.33	25.97	24.23	18.67	25.77	25.53	24.47	22.23	
		Ours	37.04	<b>38.09</b>	<b>35.94</b>	<b>34.10</b>	<b>29.57</b>	<b>37.65</b>	<b>33.33</b>	<b>31.30</b>	<b>26.22</b>		Ours	<b>37.20</b>	<b>36.53</b>	<b>36.70</b>	<b>37.10</b>	<b>36.87</b>	<b>37.40</b>	<b>37.37</b>	<b>37.27</b>	<b>36.30</b>	

Table 1: In various datasets, we compare four algorithms under two types of noise (relevant and irrelevant) and three retriever settings. The reported results represent the average performance across three different Random seed, and the best performing cases are highlighted in **bold**.

models. If a smaller model achieves performance comparable to that of a larger model, it indicates that the approach can be utilized efficiently. As shown in Table 2, the performance remains similar regardless of whether a 1.3B or a 7B model is used. This suggests that employing a smaller model as the metric model does not lead to significant performance degradation, enabling more efficient utilization. Second, we examine the impact of controlling the neighbor size  $N_{neighbor}$ , as illustrated in Figure 4. The results show that the performance remains consistent even as the number of neighbors increases. This suggests that the proposed algorithm operates efficiently even with relatively few neighbors. Based on these findings, we highlight that the proposed algorithm is both efficient and robust.

**Sensitivity of  $\gamma$ :** We examine our method’s sensitivity on the probability threshold  $\gamma$  involved in noise detection under the setting of noise ratio 0.6 using TopK retriever. As shown in Figure 5, the performance of our method remains consistent on both NQ and SciQ datasets when  $\gamma \in [0.4, 0.9]$ , with negligible drop from naive ICL of the clean setting, even the majority of the training set is noised. This shows that our method stays robust on different  $\gamma$ , and the default  $\gamma = 0.5$  can be a proper choice for various noise settings.

**Qualitative analysis of failure cases.** We conducted an in-depth analysis of failure cases on WebQ with 40% relevant noise, revealing a critical pattern: *annotation length significantly impacts detection accuracy*. We present the annotation sequence length distribution for clean/noisy samples

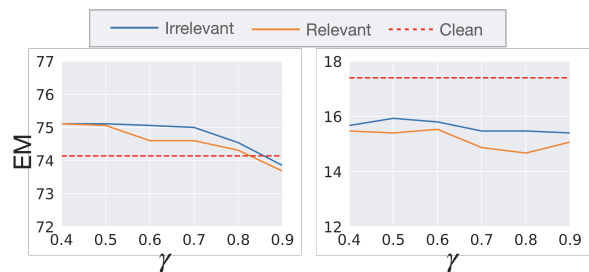


Figure 5: Sensitivity of  $\gamma$  under noise ratio 0.6 and TopK retriever. Left: SciQ; right: NQ. The ‘irrelevant’/‘relevant’ indicates the performance of our method under noised setting, and ‘clean’ indicates the performance of naive ICL baseline under clean setting.

that our method failed/successfully detected in Figure 10 of subsection B.5, respectively. Specifically: 1) Successfully detected clean samples: Only 14% had annotation lengths larger than 7 tokens; 2) Successfully detected noisy samples: Only 12% had annotation lengths larger than 7 tokens; 3) Failed detection cases (noisy misclassified as clean): 36% had annotation lengths larger than 7 tokens. This pattern is theoretically consistent with our formulation on the estimation of extrinsic bias, as longer annotations create more complex probability distributions that can obscure the distinction between clean and noisy samples. This finding provides actionable insights for future work: adaptive debiasing techniques could be designed to account for annotation length, potentially using length-normalized probability estimates or employing specialized models for longer sequences.



## 5 Related Work

**In-context learning (ICL):** Recent research has leveraged pre-trained LLMs for downstream NLP tasks through in-context learning, particularly in text classification (Yoo et al., 2022) and generation tasks (O’Brien and Lewis, 2023). Notable advances include the UDR retriever by Li et al. (2023), which works effectively across multiple tasks, and the efficient approach by Liu et al. that extracts in-context vectors from LLM embeddings to reduce computational costs. However, most ICL research assumes clean, high-quality demonstrations, leaving open questions about performance with noisy or imperfect examples.

**ICL with noisy annotations:** Initial studies exploring random labels in ICL classification have shown mixed results. While Min et al. (2022) found limited performance impact with random retrievers for certain LLM-dataset combinations, Yoo et al. (2022) demonstrated significant performance degradation across a broader range of settings. More recent work has begun addressing noisy ICL directly. Kang et al. (2024) proposed *Rectification* for classification tasks, though its fine-tuning requirements introduce substantial computational overhead. For generation tasks, Gao et al. (2024) pioneered the first noise-robust method, but it shows limitations under high-noise conditions.

**Debiasing LLM Output:** Despite their capabilities, LLMs can exhibit biases from their pre-training corpora that impact task performance. To address this, Li et al. (2022) and Zhao et al. (2024) developed Contrastive Decoding, which improves text generation quality by debiasing larger LLMs using outputs from smaller models within the same family. Additionally, Fei et al. (2023) and Zhao et al. (2021) introduced methods to reduce bias in LLMs by addressing both prefixed context bias and finite label bias in classification tasks.

## 6 Conclusion

In this paper, we have presented a robust method for handling noisy demonstrations in In-Context Learning (ICL). Our approach addresses both intrinsic and extrinsic biases in LLMs’ perception of noisy demonstrations, enabling more reliable detection of problematic examples and improving the overall inference process. Through extensive experiments across four text generation datasets, we demonstrate our method’s effectiveness under

various retrieval strategies and noise conditions, particularly showing strong performance even with noise ratios as high as 0.8. Notably, our approach achieves competitive performance without significant computational overhead, making it practical for real-world applications. We also show that our method maintains its effectiveness even when using smaller LLMs for metric computation, further enhancing its practical utility. The successful results across different settings validate not only the theoretical foundations of our dual debiasing framework but also its practical applicability. Through this work, we advance both the robust deployment of LLMs and our understanding of how these models perceive and process noisy query-annotation pairs, providing a foundation for future research in robust ICL methods.

## Limitations

While our dual debiasing approach demonstrates strong performance across multiple datasets and LLM architectures, several limitations should be acknowledged. First, our method’s effectiveness relies heavily on the quality and diversity of the neighbor samples used for extrinsic debiasing. In domains with limited available data or highly specialized knowledge, generating appropriate neighbors may be challenging, potentially affecting the accuracy of our *Sample Cleanliness Score*. Additionally, while our approach works well for text generation tasks, its applicability to other ICL applications like classification or structured prediction remains unexplored.

## Potential Risks

The computational cost of neighbor generation and metric calculation, though moderate, increases linearly with the number of demonstration samples. While we show that using fewer neighbors can maintain performance, there may be a practical upper limit to the dataset size our method can handle efficiently. Furthermore, there is a risk that our approach might inadvertently discard valid but unusual demonstrations that appear noisy due to their uniqueness, particularly in specialized domains where LLMs have limited exposure during pre-training. Users should carefully consider these limitations when applying our method to new domains or tasks.

## Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant IIS-2212174, National Institute of Aging (NIA) 1RF1AG072449, National Institute of General Medical Sciences (NIGMS) 1R01GM145700.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.
- Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. *arXiv preprint arXiv:2109.06827*.
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#). If you use this software, please cite it using these metadata.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.
- Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. *arXiv preprint arXiv:2305.19148*.
- Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. 2024. On the noise robustness of in-context learning for text generation.
- Zhizhong Huang, Junping Zhang, and Hongming Shan. 2023. Twin contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11661–11670.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Junyong Kang, Donghyun Son, Hwanjun Song, and Buru Chang. 2024. In-context learning with noisy labels. *arXiv preprint arXiv:2411.19581*.
- Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. 2022. Unicorn: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9676–9686.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.
- Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. 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.
- Sheng Liu, Haotian Ye, Lei Xing, and James Y Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Forty-first International Conference on Machine Learning*.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.
- Sean O’Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhenyu Wu Wu, Yaoxiang Wang, Jiacheng Ye, Jiangtao Feng, Jingjing Xu, Yu Qiao, and Zhiyong Wu. 2023. Openicl: An open-source framework for in-context learning. *arXiv preprint arXiv:2303.02913*.

Jianhao Yan, Pingchuan Yan, Yulong Chen, Jing Li, Xianchao Zhu, and Yue Zhang. 2024. Benchmarking gpt-4 against human translators: A comprehensive evaluation across languages, domains, and expertise levels. *arXiv preprint arXiv:2411.13775*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taek Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2437.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. *arXiv preprint arXiv:2405.02750*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

## A Experiment Setting

**Dataset** In this study, we utilize four text generation tasks. Details of the datasets are presented in Table 5. The complete ICL template for each dataset is shown in Table 4. And the noisy demonstration examples are shown in Table 3.

Dataset	Noise Ratio	gmm_part_thres	$\gamma$
NQ	[0.2, 0.8]	5.0	0.5
WebQ	[0.2, 0.8]	4.0	0.5
SCIQ	[0.2, 0.8]	10.0	0.5
SQUAD	[0.2, 0.8]	12.5	0.5

Table 6: The hyper-parameter setting for the main table.

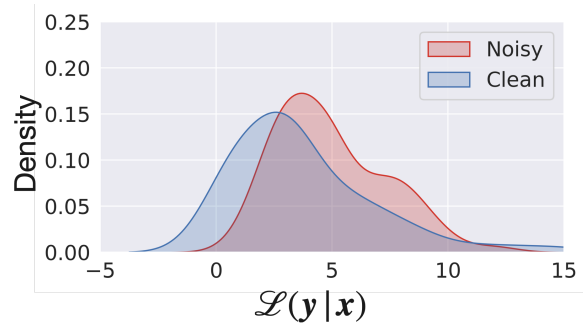


Figure 7: Distribution of  $\mathcal{L}(y|x)$  for both noisy and clean samples on NQ. Heavy overlapping between the distributions of noisy and clean samples.

**Hyper-parameter** The hyper-parameter settings for main table are shown in Table 6.

**Computation Environment** We run our experiments on NVIDIA RTX A5000 GPU. Each experiment takes less than half an hour on a single GPU.

## B More Experiment Results

### B.1 Fewer neighbors results for our method

As shown in Figure 6, fewer neighbors only lead to a negligible performance drop on our method, showing the applicability even with a very small number of neighbors to calculate the  $\{\mathcal{I}\}_{i=1}^N$ .

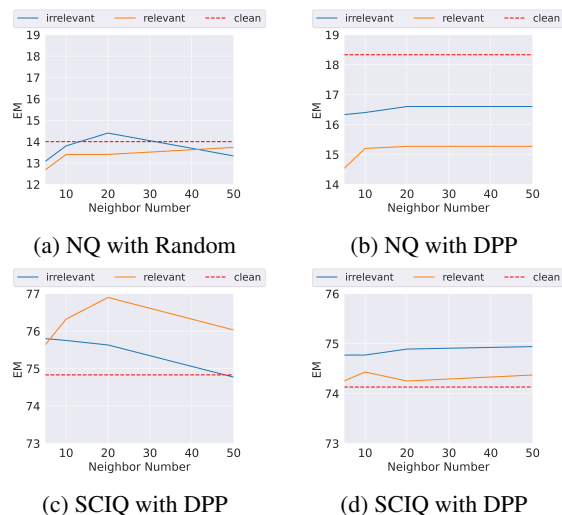


Figure 6: Performance of our method under setting of noise ratio 0.6 using different number of neighbors.

### B.2 Analysis on $\mathcal{L}(y|x)$

The visualization of  $\mathcal{L}(y|x)$  on NQ is shown in Figure 7. It shows that using  $\mathcal{L}(y|x)$  directly is not sufficient to distinguish clean and noisy samples.

Dataset	Setting	In-Context Demonstration
NQ	Clean	<b>Question:</b> how i.met your mother who is the mother? <b>Answer:</b> Tracy McConnell
	Irrelevant	<b>Question:</b> how i.met your mother who is the mother? <b>Answer:</b> Moreirense F.C.
	Relevant	<b>Question:</b> how i.met your mother who is the mother? <b>Answer:</b> Barney Stinson is the mother
WebQ	Clean	<b>Question:</b> where are the nfl redskins from? <b>Answer:</b> Washington Redskins
	Irrelevant	<b>Question:</b> where are the nfl redskins from? <b>Answer:</b> the Bee Gees
	Relevant	<b>Question:</b> where are the nfl redskins from? <b>Answer:</b> Los Angeles, California
SCIQ	Clean	<b>Support:</b> It might only take one sperm to fertilize an egg, but that sperm is not alone. Hundreds of millions of sperm can be released during sexual intercourse. <b>Question:</b> How many sperm does it take to fertilize an egg? <b>Answer:</b> 1
	Irrelevant	<b>Support:</b> It might only take one sperm to fertilize an egg, but that sperm is not alone. Hundreds of millions of sperm can be released during sexual intercourse. <b>Question:</b> How many sperm does it take to fertilize an egg? <b>Answer:</b> open clusters
	Relevant	<b>Support:</b> It might only take one sperm to fertilize an egg, but that sperm is not alone. Hundreds of millions of sperm can be released during sexual intercourse. <b>Question:</b> How many sperm does it take to fertilize an egg? <b>Answer:</b> 3
SQuAD	Clean	<b>Question:</b> What was the name of the streaming service? <b>Context:</b> On February 6, 2016, one day before her performance at the Super Bowl, Beyoncé released a new single exclusively on music streaming service Tidal called "Formation". <b>Answer:</b> Tidal
	Irrelevant	<b>Question:</b> What was the name of the streaming service? <b>Context:</b> On February 6, 2016, one day before her performance at the Super Bowl, Beyoncé released a new single exclusively on music streaming service Tidal called "Formation". <b>Answer:</b> village
	Relevant	<b>Question:</b> What was the name of the streaming service? <b>Context:</b> On February 6, 2016, one day before her performance at the Super Bowl, Beyoncé released a new single exclusively on music streaming service Tidal called "Formation". <b>Answer:</b> Spotify

Table 3: Clean/noisy demonstration examples for each dataset.

Dataset	Prompt	Example
NQ	<b>Question:</b> <Question>	<b>Question:</b> what do the 3 dots mean in math
	<b>Answer:</b> <Answer>	<b>Answer:</b> the therefore sign
WebQ	<b>Question:</b> <Question>	<b>Question:</b> what is the oregon ducks 2012 football schedule?
	<b>Answer:</b> <Answer>	<b>Answer:</b> University of Oregon
SCIQ	<b>Support:</b> <Support>	<b>Support:</b> Smooth muscle regulates air flow in lungs.
	<b>Question:</b> <Question> <b>Answer:</b> <Answer>	<b>Question:</b> Which kind of muscle regulates air flow in lungs? <b>Answer:</b> smooth
SQuAD	<b>Question:</b> <Question>	<b>Question:</b> Who won the Super Bowl MVP?
	<b>Context:</b> <Context>	<b>Context:</b> The Broncos took an early lead in Super Bowl 50 and never trailed. Newton was limited by Denver’s defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown. Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, $2\frac{1}{2}$ sacks, and two forced fumbles.
	<b>Answer:</b> <Answer>	<b>Answer:</b> Von Miller

Table 4: The ICL template for datasets. Placeholders (e.g., <Question> and <Answer>) will be replaced by real questions or answers.

### B.3 Robustness on corpus

the result on SCIQ.

We conduct the comparison experiment between using  $\mathcal{C}_{in}$  and  $\mathcal{C}_{out}$  under noise ratio 0.6. Figure 8 presents the result on NQ, and Figure 9 presents

Dataset	Task	Training	Test
NQ	Open-Domain QA	10,000	500
WebQ	Open-Domain QA	1,261	1,213
SCIQ	Reading Comprehension	6,059	580
SQuAD	Reading Comprehension	20,000	1,000

Table 5: The statistics of the datasets used.

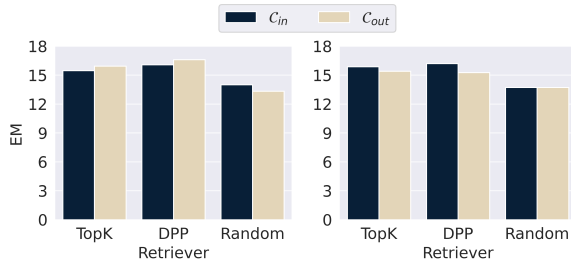


Figure 8: Comparison between using  $C_{in}$  and  $C_{out}$  on NQ under noise ratio 0.6. Left: irrelevant noise; Right: relevant noise.

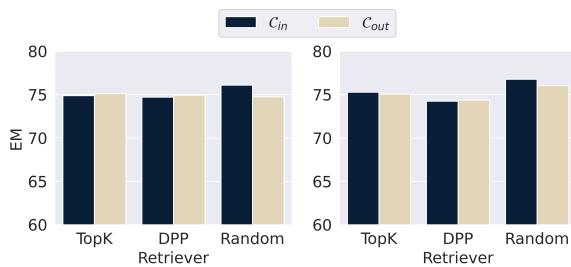


Figure 9: Comparison between using  $C_{in}$  and  $C_{out}$  on SCIQ under noise ratio 0.6. Left: irrelevant noise; Right: relevant noise.

#### B.4 Dependency on choice of LLMs

We conduct further analysis on how different LLMs benefit from our dual-debiasing approach under 40% relevant noise setting on NQ. And we present the AUC improvement of noisy sample detection using our method  $\Delta AUC = AUC_{ours} - AUC_{naive}$  in Table 7, where  $AUC_{naive}$  is the AUC of using naive per-token loss  $\mathcal{L}(\mathbf{y}|\mathbf{x})$ . This reveals an important insight: while our method improves performance across all models, smaller LLMs like GPT-Neo-1.3B demonstrate substantially larger gains. This suggests our approach is particularly valuable for resource-constrained scenarios, effectively democratizing robust ICL capabilities across model scales. The consistency of improvement across architectures also demonstrates the generalizability of our dual-debiasing framework.

LLM	$\Delta AUC \uparrow$
GPT-Neo-1.3B	0.2157
Gemma-2B	0.1403
Mistral-7B-v0.1	0.1179
Llama-2-7B	0.1476

Table 7: The improvement of noise detection AUC using our method across various LLMs on NQ with 0.4 relevant noise.

#### B.5 Results for analysis of failure cases

We conducted an in-depth analysis of failure cases on WebQ with 40% relevant noise using Llama-2-7B. We present the annotation sequence length distribution for clean/noisy samples that our method failed/successfully detected in Figure 10, respectively.

### C Pseudocode of our method

We provide the complete pseudocode for our noisy ICL framework in algorithm 1.

### D Generation of neighboring samples

Our neighbor generation process follows a principled approach as described below: For each demonstration sample  $(\mathbf{x}, \tilde{\mathbf{y}})$ , we construct  $N_{neighbor}$  samples by pairing the original query  $\mathbf{x}$  with randomly sampled annotations from a given corpus  $\mathcal{C}$ . Notice that we will drop the annotation sampled from  $\mathcal{C}$  if its tokenized sequence length is larger than the neighbor radius  $\eta$  defined for  $\mathcal{N}_{DISC}$ . The quality of these neighbors is rigorously controlled through the Edit Distance metric, which quantifies the textual difference between annotations.

We enforce a maximum distance constraint  $\eta = \max\{T, T_{max}\}$ , where  $T$  is the observed annotation length and  $T_{max}$  is the maximum annotation sequence length of the observed annotation from  $\tilde{\mathcal{D}}_{train}$ . This formal constraint ensures that neighbors remain within a bounded semantic radius of the original sample, maintaining relevance while providing sufficient diversity for robust debiasing.

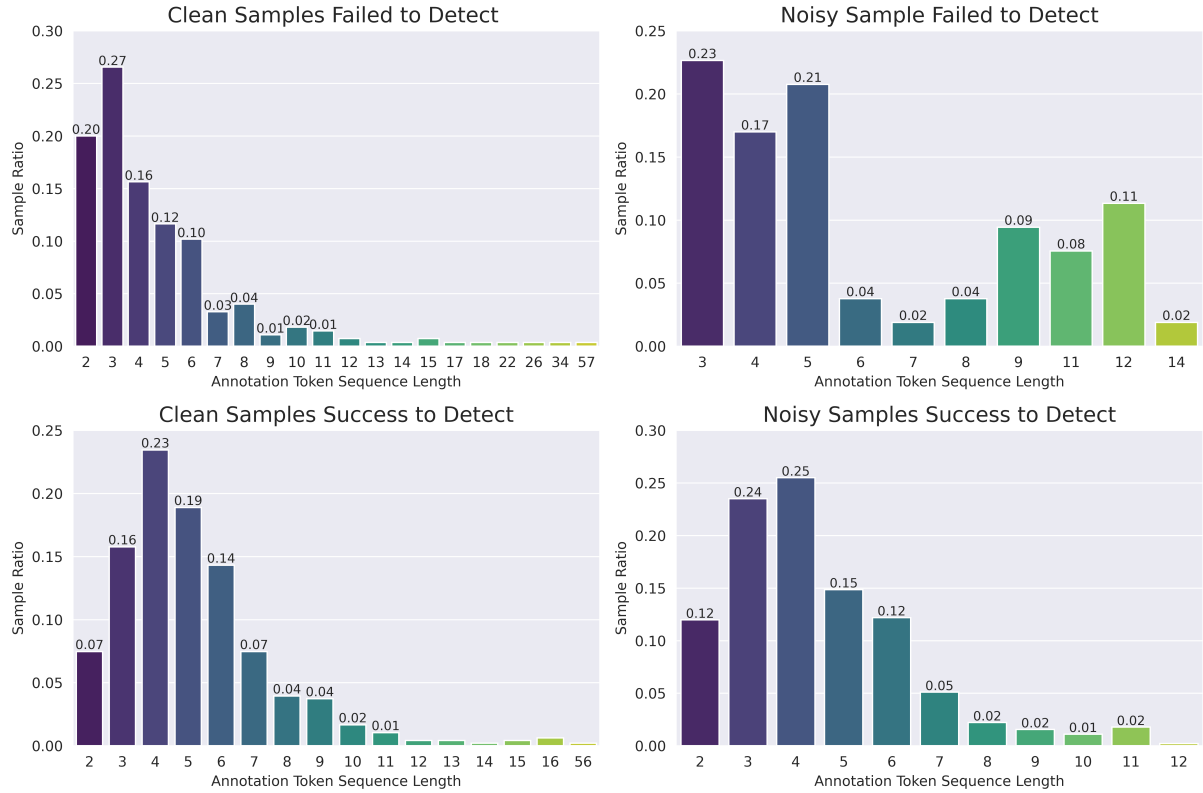


Figure 10: Annotation sequence length distribution for clean/noisy samples that our method failed/successfully detected under WebQ with 40% relevant noise using Llama-2-7B.

---

**Algorithm 1** Complete pseudocode for proposed dual-debiasing framework for noisy ICL

---

**Require:** LLM model  $\mathcal{M}$ , observed training set  $\tilde{\mathcal{D}}_{\text{train}}$  with noisy demonstration samples, large corpus  $\mathcal{C}$ .

*/\* Metric Calculation For the Whole Training Set \*/*

- 1: Initialize empty metric score set  $U = \emptyset$
  - 2: **for** each demonstration example  $(\mathbf{x}_i, \tilde{\mathbf{y}}_i) \in \tilde{\mathcal{D}}_{\text{train}}$  **do**
  - 3:   Calculate  $\mathcal{L}_{\text{de-int}}(\tilde{\mathbf{y}}|\mathbf{x})$  based on Equation 4
  - 4:   Construct neighbors  $\mathcal{N}_{\text{DISC}}((\mathbf{x}_i, \tilde{\mathbf{y}}_i))$  using  $\mathcal{C}$  as described in subsection 3.3
  - 5:   Calculate Sample Cleanliness Score  $\mathcal{I}_i$  using  $\mathcal{N}_{\text{DISC}}((\mathbf{x}_i, \tilde{\mathbf{y}}_i))$  based on Equation 8 and Equation 10
  - 6:   Add  $\mathcal{I}_i$  to  $U$
  - 7: **end for**
  - /\* Regular ICL on clean train subset \*/*
  - 8: Perform GMM-based noisy sample detection on score set  $U = \{\mathcal{I}_i\}_{i=1}^N$  as described in subsection 3.4
  - 9: Separate training set into clean subset  $\tilde{\mathcal{D}}^{\text{clean}}$  and noisy subset  $\tilde{\mathcal{D}}^{\text{noisy}}$
  - 10: Remove noisy subset, keeping  $\tilde{\mathcal{D}}'_{\text{train}} = \tilde{\mathcal{D}}^{\text{clean}}$
  - /\* Metric Calculation For the Whole Training Set \*/*
  - 11: For each test query  $\mathbf{x}_j^{\text{test}} \in \mathcal{D}_{\text{test}}$ , perform regular ICL using  $\tilde{\mathcal{D}}'_{\text{train}}$  as retrieval pool
-