In-context Learning Demonstration Generation with Text Distillation

Wuyuqing Wang, **Erkun Yang***, **Zilan Zhou** and **Cheng Deng**
Xidian University, Xi'an, China
24021211899@stu.xidian.edu.cn, erkunyang@gmail.com,
24021211818@stu.xidian.edu.cn, chdeng.xd@gmail.com

Abstract

In-context learning (ICL), a paradigm derived from large language models (LLMs), holds significant promise but is notably sensitive to the choice of input demonstrations. While numerous methodologies have been developed to select the optimal demonstrations from existing datasets, our work alternatively proposes to generate representative demonstrations through a Distillation-based Demonstration Generation (DDG) framework. Specifically, our approach aims to generate demonstrations that encapsulate the essential attributes of the target dataset. Rather than optimizing these demonstrations directly, we design a generative model and try to refine it by minimizing the discrepancies between the calculative models trained on generated demonstrations and the original datasets respectively. Additionally, we leverage a teacher-student framework to stabilize the training process and improve the quality of the synthesized samples. Extensive experiments conducted across ten prevalent text datasets demonstrate that our DDG method substantially outperforms existing state-of-the-art methodologies. Our code will be available at https://github.com/wwyq1/DDG.

1 Introduction

In-Context Learning (ICL) has risen as an influential approach for applying large language models (LLMs) to address new tasks during inference [Dong *et al.*, 2024]. ICL enables a model to adjust to various tasks without the need for further training, depending solely on the given prompt, unlike traditional methods that necessitate task-specific fine-tuning. This adaptability not only diminishes the costs associated with adapting to new tasks but also provides a clear and adaptable method for steering the model's actions [S. *et al.*, 2024]. Utilizing the demonstrations in the prompt, ICL enhances generalization over a broad spectrum of tasks and improves the reasoning ability of LLMs [Dong *et al.*, 2024].

Nevertheless, the effectiveness of ICL heavily relies on the demonstrations contained in the prompt, where minor changes in these demonstrations can drastically affect the model's performance [Dong *et al.*, 2024]. To address this limitation, many different approaches have been proposed for demonstrations selection, e.g., selecting demonstrations which are similar to the query sample in the embedding space [Liu *et al.*, 2022; Wu *et al.*, 2023], learning a deep learning-based demonstrations retriever [Luo *et al.*, 2024; Li and Qiu, 2023], selecting demonstrations based on LLM feedback [Wang *et al.*, 2023; Chen *et al.*, 2023; Liu *et al.*, 2024a] or influence analysis [Nguyen and Wong, 2023; S. *et al.*, 2024]. However, those selection-based methods always discard a large fraction of unselected samples, dismissing their contribution to ICL and often resulting in sub-optimal performance. Moreover, many of these methodologies are tailored for specific LLMs. And the selected demonstrations cannot generalize well to other LLMs.

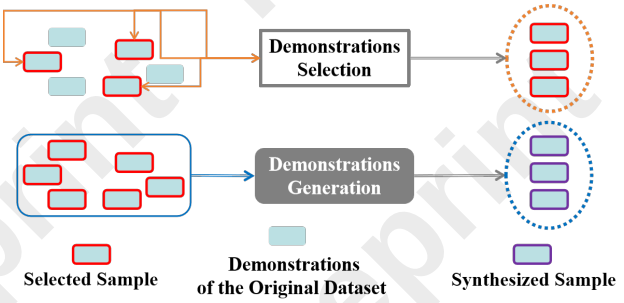

Figure 1: Comparison of the principles between demonstrations selection approach and demonstrations generation approach

To address the challenges outlined above, we propose a novel method for generating more representative demonstrations that encapsulate the essential information of the entire training dataset. Specifically, we introduce a **Distillation-based Demonstrations Generation (DDG)** framework, consisting of two key components: the generative model and the calculative models. The generative model is responsible for generating demonstrations, while the calculative models are tasked with ensuring that these generated demonstrations are as representative as possible of the original dataset. Inspired by existing data distillation techniques, we frame our objective as a minimization problem. Specifically, we aim to minimize the discrepancy between the gradients of the calculative models' parameters during the gradient-descent-based opti-

*Corresponding author.

mization process, computed over two distinct sets: one set derived from the original training dataset and the other from the generated demonstrations. However, most prior data distillation methodologies are designed for continuous image data, which is not directly applicable to text with discrete representations. To overcome this, we integrate the generative model with the calculative models and alternately optimize the parameters of both components, ensuring the generative model learns to generate highly representative demonstrations. Furthermore, to enhance the learning stability and improve the performance of the calculative models, we adopt a teacher-student framework. Through this combined approach, we aim to improve the efficiency and quality of the generated demonstrations, ultimately enabling more effective ICL.

We assess the efficacy of DDG on ten widely utilized text classification datasets, including eight short-text datasets: SST-2, SST-5, MNLI, QQP, CoLA, AGNews, QNLI, and CR; in addition to two complex long-text multi-tag datasets: BANKING77 and GoEmotions. Utilizing the optimally synthesized samples, we conduct experiments in conjunction with four prominent LLMs (LLaMA-2, LongLLaMA, Qwen, and Mistral) to evaluate the performance of ICL. Relative to the baseline methodologies, under the same conditions, the classification accuracies of DDG will generally increase by 7% on average for short-text datasets, and 5% on average for long-text multi-tag datasets.

To sum up, our contributions are as follows:

- Rather than selecting representative demonstrations from existing datasets, we propose a pioneering approach using a distillation-based demonstrations generation framework to synthesize more informative samples, which is among the earliest attempts to employ data distillation techniques for the ICL task.
- Instead of optimizing synthesized samples directly, we develop a generative model and incorporate it with the calculative models to synthesize completely new samples at each iteration. Moreover, a teacher-student framework is also employed, which can further improve the stability of the training process.
- Extensive experiments on ten commonly used text datasets show that our proposed approach can significantly outperform existing state-of-the-art methodologies for the ICL task.

2 Related Work

2.1 In-Context Learning

As model and data sizes scale, large language models (LLMs) exhibit in-context learning (ICL) ability, learning from a few natural language template-based demonstrations [Dong *et al.*, 2024]. However, ICL performance is often unstable, and highly sensitive to prompt configuration, including demonstrations selection, formatting, and ordering [Lu *et al.*, 2022a; Rubin *et al.*, 2022]. Consequently, various demonstrations selection methodologies have been explored, including heuristic strategies [Peng *et al.*, 2024; Liu *et al.*, 2024a]; retrievers trained using in-batch negative loss [Li *et al.*, 2023] or reinforcement learning [Scarlatos and Lan, 2024]; and LLM-feedback based methods that leverage prediction confidence [Wang *et al.*, 2023; S. *et al.*, 2024; Scarlatos and Lan, 2024], these latter methods can also be considered influence-based, analyzing the impact of training samples using LLMs. [Lu *et al.*, 2022b] suggests that the ordering of demonstrations can be optimized for performance gain, LLMs have shown a tendency to overly rely on the most frequent labels or labels that appear at late positions in the prompt [Liu *et al.*, 2024c]. Another research trend involves utilizing LLMs to reformat the representation of existing demonstrations [Yang *et al.*, 2024; Liu *et al.*, 2024b], thereby enhancing the model's ability to follow the demonstrations more effectively.

2.2 Data Distillation

Data distillation aims to create a compact representation of the original dataset while preserving its core information [Wang *et al.*, 2018; Yang *et al.*, 2019]. Current researches on data distillation primarily focus on image datasets due to their continuous nature, with various high-quality distillation methodologies proposed: Meta-model matching methodologies solve the original bi-level optimization formulation such as DC [Zhao and Bilen, 2021]; DM [Zhao and Bilen, 2023] seeks to minimize the statistical distance between real and distilled samples; TESLA [Cui *et al.*, 2023] optimized synthetic samples to approximate trajectories of model parameters trained with real data; [Qin *et al.*, 2024] introduced learnable soft-labels, which are optimized together with input images to make each synthetic sample more informative. For textual datasets, the discrete nature of text poses challenges, yet recent innovations have emerged. For instance, studies by [Maekawa *et al.*, 2023; Li *et al.*, 2024] map discrete text samples into continuous word embedding vectors. However, these synthetic datasets are incompatible with models using different embedding weights, [Maekawa *et al.*, 2024] introduced a approach that synthesizes datasets unsteadily by optimizing the continuous parameters of a generator model.

3 Proposed Method

Current large language models (LLMs), including GPT-3 and LLaMA-2, have demonstrated excellent in-context learning (ICL) capabilities. Given an original dataset $D_{org} = \{x_1, \ldots, x_N\}$ with N training samples, ICL is designed to enable LLMs to learn from the prompt containing task instruction, demonstrations, query directly, without additional training of model parameters as opposed to prompt learning, few-shot learning and so on [Dong *et al.*, 2024]. Therefore, the performance of ICL will mainly depend on the demonstrations inputted to LLMs. However, although existing methodologies of demonstrations selection already have excellent ability to optimize the prompt, they always ignore the valuable information contained in the unselected samples.

Inspired by [Zhao and Bilen, 2021] and [Maekawa *et al.*, 2024], we propose a novel approach DDG to generate more represesntative demonstrations $D_{syn} = \{\bar{x}_1, \ldots, \bar{x}_M\} (M \ll N)$ relative to the original dataset through training the generative model with data distillation techniques, meanwhile, we adopt a teacher-student framework to stabilize and improve the training process.

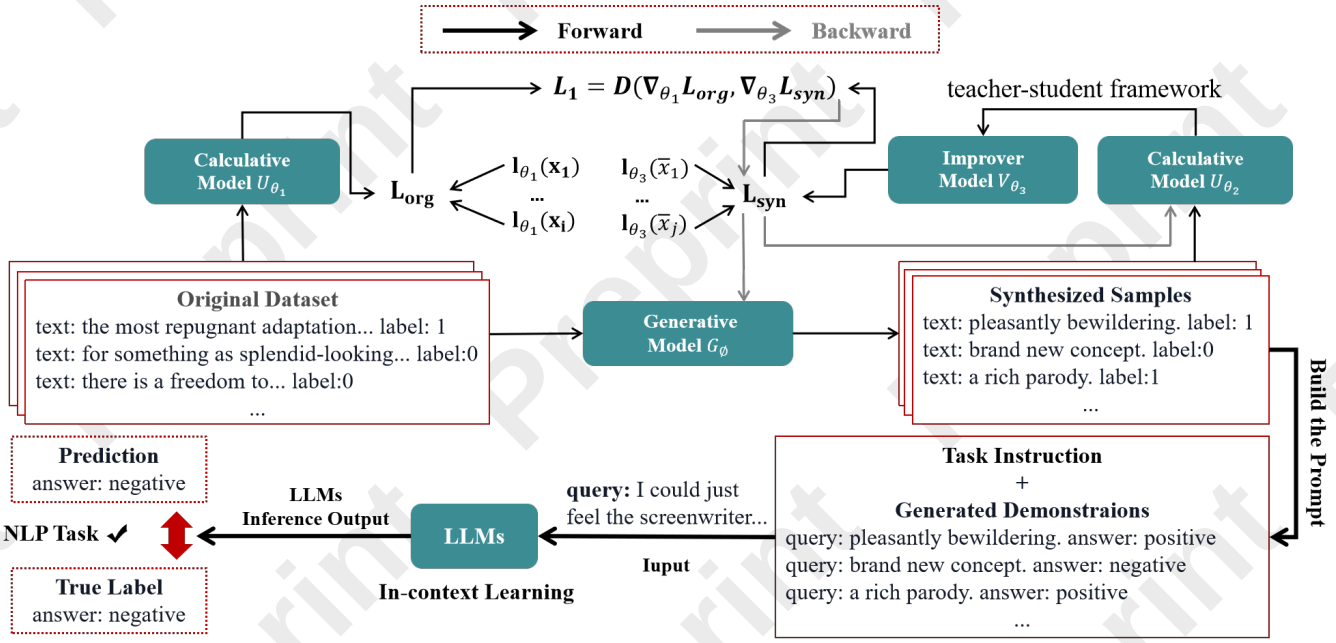

Figure 2: This is the flow of our method DDG. Initially, the losses of the original dataset and the generated demonstrations are calculated based on the calculative models respectively, and then, the generative model G_ϕ is optimized iteratively by the gradient matching loss based on the calculative model U_{θ_1} and the improver model V_{θ_3} combined with teacher-student framework to generate more representative demonstrations for the performance enhancement of ICL.

3.1 Distillation-based Demonstration Generation

We define the calculative models that could correctly predict the label of previously unseen text, setting the calculative models with two different variants: one U_{θ_1} trained on the original dataset with parameter θ_1, and the U_{θ_2} trained on the generated demonstrations with parameter θ_2. In words, we wish to optimize the generative model G_ϕ with parameter ϕ to synthesize distilled samples such that U_{θ_1} achieves not only comparable generalization performance to U_{θ_2} but also converges to a similar solution in the same parameter space:

$$minD\left(\theta_1, \theta_2\right), \tag{1}$$

where the D function is a cosine similarity-based distance function, which is expressed as:

$$D(\alpha, \beta) = 1 - \frac{\alpha \cdot \beta}{\|\alpha\|\|\beta\|}. \tag{2}$$

Following [Zhao and Bilen, 2021], we introduce the gradient matching loss L_1, which not only ensures that the parameters of both calculative model variants are optimized to match as closely as possible in each iteration with similar updating paths, but is also used to update the parameter ϕ as follows:

$$L_1 = \sum_{r=1}^{E} D\left(\nabla_{\theta_1} L_{\text{org}}, \nabla_{\theta_2} L_{\text{syn}}\right), \tag{3}$$

where E is the total number of iterations, L_{org} and L_{syn} are the loss based on the original dataset and generated demonstrations, respectively.

For the original dataset D_{org} and the calculative model U_{θ_1}, we design the loss L_{org}:

$$L_{\text{org}} = \frac{1}{N} \sum_{i=1}^{N} l(U_{\theta_1}(x_i)), \tag{4}$$

where the l function represents the cross-entropy-based loss.

Meanwhile, for generated demonstrations and the calculative model U_{θ_2}, we design the loss L_{syn}. Due to the discrete nature of textual samples, it is not feasible to directly apply the back-propagation process based on gradient-descent. When computing the loss, instead of simply averaging the losses for all synthesized samples, inspired by the [Maekawa *et al.*, 2024], we draw on the work of [Hiraoka *et al.*, 2020] to design back-propagation process. Therefore, L_{syn} can be back-propagated to G_ϕ through the differentiable pass via weights μ_j and generation probabilities $P(G_\phi(x_i) \Rightarrow \bar{x}_j)$:

$$L_{\text{syn}} = \sum_{j=1}^{M} \mu_j \, l(U_{\theta_2}(\bar{x}_j)), \tag{5}$$

where

$$\mu_j = \frac{P(G_\phi(x_i) \Rightarrow \bar{x}_j)}{\sum_{q=1}^{M} P(G_\phi(x_i) \Rightarrow \bar{x}_q)}. \tag{6}$$

Teacher-Student Framework

To optimize the training process of U_{θ_2} based on generated demonstrations, we utilize the teacher-student framework [Abbasi *et al.*, 2020], designating the calculative model

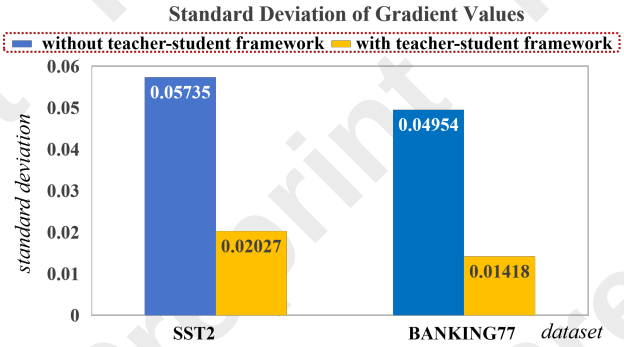

Figure 3: Standard deviation of model's gradient values during the training process with or without teacher-student framework on different datasets.

U_{θ_2} as the teacher model and the improver model V_{θ_3} as the student model. As illustrated in Figure 2, we anticipate that the student model V_{θ_3} can quickly learn from the teacher model U_{θ_2}, thereby achieving relatively superior performance while minimizing training costs.

Therefore, we utilize the exponential moving average (EMA) fitting under the teacher-student framework. In DDG, the student model V_{θ_3} based on the teacher model U_{θ_2} can be defined using the λ parameter $(0 \leq \lambda \leq 1)$ as follows:

$$\theta_3 = \lambda * \theta_3 + (1 - \lambda) * \theta_2, \tag{7}$$

$\lambda = 0.99$ was selected empirically for this paper. During subsequent training, we replace θ_2 with θ_3 in equation (3), U_{θ_2} with V_{θ_3} in equation (5).

In addition, the cross-entropy loss function l is enhanced through the EMA fitting. It is well established that in the cross-entropy loss function, the $shift\_logit$ parameter typically represents the probability distribution of the relevant parameters [Mao *et al.*, 2023], so we combined it with the λ parameter to define the new core factor A_logit:

$$A\_logits = \lambda * A\_logits + (1 - \lambda) * shift\_logit. \tag{8}$$

Furthermore, to rigorously evaluate the validity of the teacher-student framework, we independently calculated the standard deviation of gradient values in the final layer for models trained on generated demonstrations under two conditions: with and without the framework during training. As illustrated in Figure 3, the integration of the teacher-student framework results in a substantial reduction in the standard deviation of gradient values. This attenuation directly correlates with diminished parameter update fluctuations during model training, attributable to EMA-based parameter smoothing. Consequently, these empirical observations demonstrate that the teacher-student framework significantly enhances the stability of the student model V_{θ_3} throughout the training process, and facilitates more efficient convergence of the generative model G_ϕ's iteration by stabilizing parameter optimization trajectories.

3.2 Algorithm for Gradient Matching

We apply the pre-trained LLM G_{ϕ_0} combined with the language modelling loss as the basis for the generative model G_ϕ. Therefore, we design a gradient-descent-based matching algorithm for fine-tuning of the generative model's continuous parameter ϕ. As demonstrated in Algorithm 1, we have devised a nested loop algorithm for gradient matching to iteratively optimize the model parameters. This algorithm comprises an outer-loop dedicated to the initialization of the parameter θ_0 to enhance the adaptation of DDG to previously unseen models, and an inner-loop tasked with computing the gradient matching loss L_1 for each class. Furthermore, in nested inner-loop, We also designed three parameter update processes: (1) the steps to optimize the parameter ϕ of the generative model G_ϕ based on the AdamW optimizer, (2) the steps for the calculative model U_{θ_2} to optimize the parameter θ_2 with the AdamW optimizer, and (3) the steps for updating the parameter θ_3 of the improver model V_{θ_3}, under the teacher-student framework combined with EMA fitting.

Algorithm 1 The training process of DDG

Relevant parameters: ϕ: generative model parameter; θ_1, θ_2: calculative models parameter; θ_3: improver model parameter; C: the number of data classes; G_ϕ: generative model; OL: the number of outer-loop steps; IL: the number of inner-loop steps; ε: the number of steps for updating ϕ; τ: the number of steps for updating θ; φ: learning rate of ϕ; ω: learning rate of θ; λ: EMA fitting parameter.
for $ol = 1, \ldots, OL$ **do**
  \\ Outer-loop
  Initialize parameter θ_0 and ϕ_0 according to the pre-trained modals
  for $il = 1, \ldots, IL$ **do**
    \\ Inner-loop
    for $c = 1, \ldots, C$ **do**
      \\Calculated for each class of the original dataset
      $L^c_{\text{org}} = \frac{1}{N_c} \sum_{i=1}^{N_c} l_{\theta_1}(x_i)^c$
      $\bar{x}_j (j = 1 \ldots M_c) \Leftarrow G_\phi(x_i)(i = 1 \ldots N_c)$
      $L^c_{\text{syn}} = \sum_{j=1}^{M_c} \mu_j \, l_{\theta_3}(\bar{x}_j)^c$
      $L^c_1 \leftarrow D\left(\nabla_{\theta_1} L^c_{\text{org}}, \nabla_{\theta_3} L^c_{\text{syn}}\right)$
    end for
    optimization of ϕ based on ϕ_0 and $\frac{1}{C} \sum_{c=1}^{C} L^c_1$, combined with the AdamW optimizer, parameters φ and ε.
    update of θ_2 based on θ_0 and $\sum_{c=1}^{C} L^c_{\text{syn}}$, combined with the AdamW optimizer, parameters ω and τ.
    $\theta_3 = \lambda * \theta_3 + (1 - \lambda) * \theta_2, \quad (0 \leq \lambda \leq 1)$
    \\ update of θ_3 based on teacher-student framework combined with EMA fitting
  end for
end for

3.3 Synthesized samples generation

During the whole generation process, the LLMs are typically employed to revisit the entire sequence of tokens to forecast the next token [Dhamala *et al.*, 2023]. Therefore, we adopt an innovative approach that combines top-k sampling with top-p sampling, along with the temperature parameter, which aims to generate demonstrations with higher informativeness.

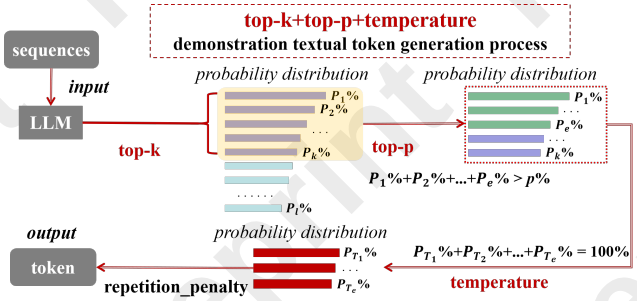

Figure 4: Illustration of textual token generation.

As illustrated in Figure 4, we first employ the top-k method to select the k tokens $w_1, \ldots, w_k$ in descending order based on the generation probability distribution $P_1\%, P_2\%, \ldots, P_l\%$ of the candidate tokens. Next, we apply a dynamic token candidate list sizing strategy known as top-p approach to accumulate the generation probabilities and select the tokens whose cumulative sum exceeds $p\%$:

$$P_1\% + P_2\% \cdots + P_e\% \geq p\%. \tag{9}$$

Again, we introduce the temperature parameter T to smooth the candidate tokens' generation probabilities distribution:

$$P_T(w_t) = \frac{exp(w_t/T)}{\sum_{r=1}^{e} exp(w_r/T)}. \tag{10}$$

Therefore, we obtain P_T from the Softmax output as $P_{T_1}\%, P_{T_2}\%, \ldots, P_{T_e}\%$, which facilitates balancing accuracy and diversity across the whole generation process, and:

$$P_{T_1}\% + P_{T_2}\% \cdots + P_{T_e}\% = 100\%. \tag{11}$$

Finally, we also use the $repetition\_penalty$ parameter specific to LLMs, which aims to reduce text repetition by reducing the generation probabilities of already synthesized tokens.

In summary, we innovatively combine three textual token generation approaches and one parameter of the LLMs [Keskar *et al.*, 2019], which in turn overcome the problems of monotonous textual tokens generation, object duplication, and unstable text quality in previous studies [Nguyen *et al.*, 2025; Dhamala *et al.*, 2023; Basu *et al.*, 2021] while balancing the trade-off between quality and diversity.

4 Experiment

4.1 Datasets

We utilized eight commonly used short-text datasets, including four distinct categories: Semantic Analysis (SST2, SST5, CR, COLA), Natural Language Reasoning (MNLI, QNLI), Text Summarisation (AGNews), and Paragraph Detection (QQP), and also two long-text multi-tag datasets: intent classification (BANKING77) and fine-grained sentiment classification (GoEmotions).

4.2 Baselines

For short-text datasets, we compared DDG with prior competitive ICL methodologies using LLaMA-series as inference LLM, including: random, BM25 [Robertson and Zaragoza, 2009], RICES [Yang *et al.*, 2022], TopK [Liu *et al.*, 2022], TopK + MDL [Wu *et al.*, 2023], GC [Jiang *et al.*, 2023], InfICL [S. *et al.*, 2024], TopK + ConE [Peng *et al.*, 2024], DILM [Maekawa *et al.*, 2024]. For long-text multi-tag datasets, We compared DDG with baseline methodology LongICLBench [Li *et al.*, 2025].

4.3 Implementation Detaills

In-Context Learning (ICL)

In this paper, we employed LLMs for classification tasks to evaluate the performance of ICL based on the generated demonstrations by DDG, and utilized the classification accuracies as evaluation metric. Initially, we employed the LLaMA-2-7B model aligned with the experimental settings of prior similar works uniformly for ICL, ensuring comparability and continuity with existing researches. Subsequently, we configured the demonstrations within the prompt to a 5-shot format across short-text datasets through the sampling process, where each shot corresponds to the random selection of one synthesized sample from each class. To evaluate the classification accuracies of ICL, we extracted 50 samples from the test set as quary, ensuring a balanced distribution of label types. Following experiments before, we also examined the efficacy of DDG in processing long-text multi-tag datasets. In ICL, we analyzed classification accuracies regarding the baseline methodology across various token length constraint. The demonstrations in the prompt were varied from 1 round (1R) to 10 rounds (10R), with each round (R) representing the random selection of one synthesized sample from each class. For testing purposes, we extracted 500 samples from the test set as quary, also ensuring a balanced distribution of label types. The rest of the experimental setup remained consistent with the aforementioned procedures. Additionally, four LLMs were selected for ICL tasks: LLaMA-2-7B, Qwen-1.5-7B, Mistral-7B, and Long-LLaMA.

The Training Process of DDG

In this paper, the GPT-3 model is selected as the basis of the generative model G_ϕ, while the RoBERTa-large model is chosen as the pre-trained model for the calculative models. We set the parameters of the nested loop algorithm for training the optimal generative model as follows: the total number of training sessions for the initial training with language modeling loss functions [Kaplan *et al.*, 2020] is 50,000, and the total number of training sessions for fine-tuning the parameters of the generative model G_ϕ is 10,000. The number of inner-loop steps is set to $IL = 50$, and the number of outer-loop steps is calculated as OL = total number of training sessions / number of inner-loop steps. The learning rate is established at 1.0×10^{-4}, and the number of updating steps ε is set to 100. The mini-batch sizes for original and synthesized samples are set to N = 200 and M = 50. The warmup ratio for the entire process is set to 0.05, weight decay is set to 0.01, gradient clipping is set to 1.0, and the dropout ratio is set to 0.1. Finally, the generative model G_ϕ was set to synthesize five samples simultaneously for each iteration, and each sample was generated with strict reference to the trainers' setting. Moreover, we trained the calculative model parameters θ_1 and θ_2 separately on the original datasets and generated

Methods	SST2	MNLI	COLA	AGNews	QNLI	CR	SST5	QQP
random	94.4	51.0	-	83.5	56.2	92.3	50.4	-
BM25 [Robertson and Zaragoza, 2009]	94.5	57.0	-	92.5	59.0	92.8	52.6	-
RICES [Yang *et al.*, 2022]	93.9	-	73.7	-	-	-	-	-
TopK [Liu *et al.*, 2022]	95.2	57.8	-	92.4	61.3	92.8	52.6	-
TopK+MDL [Wu *et al.*, 2023]	95.1	57.9	-	92.3	64.5	93.4	52.7	-
GC [Jiang *et al.*, 2023]	95.7	-	-	87.8	-	92.3	47.4	65.1
InfICL [S. *et al.*, 2024]	95.2	-	74.8	-	-	-	-	-
TopK + ConE [Peng *et al.*, 2024]	95.4	59.5	-	92.8	66.4	93.1	52.5	-
DILM [Maekawa *et al.*, 2024]	95.1	-	-	-	-	-	-	-
DDG (ours)	**97.3**	**64.0**	**82.0**	**93.3**	**83.3**	**94.7**	**54.7**	**76.7**

Table 1: Classification accuracies (%) of ICL for eight commonly used short-text datasets. DDG results are the average of the best three experiments.

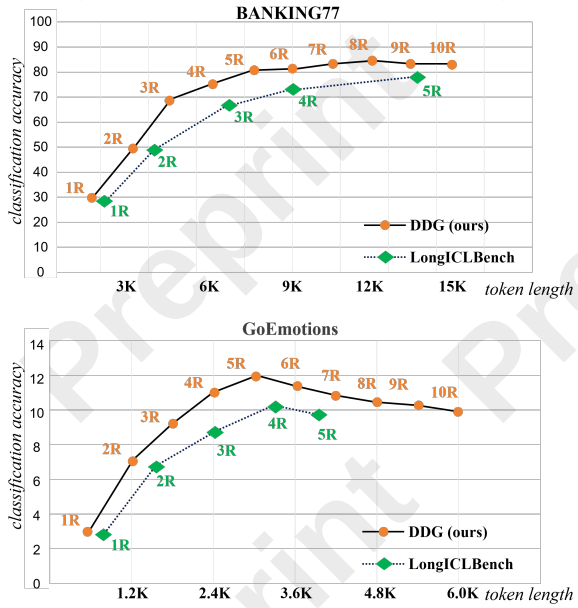

Figure 5: Average classification accuracies (%) based on different LLMs of ICL for DDG and LongICLBench under the same round (R) settings for BANKING77 and GoEmotions datasets, DDG results are the average of the best three experiments.

demonstrations five times in a loop with the learning rate of 1.0×10^{-3}, the number of updating steps τ is set to 10. Simultaneously, the λ parameter of 0.99 was chosen to update the improver model V_{θ_3} under the teacher-student framework. For short-text datasets such as SST2, we empirically set k=5-8, p=0.97, temperature to 0.7, $repetition\_penalty$ parameter value to 1.2; for long-text multi-tag datasets such as BANKING77, it is more appropriate to set k=10-12, p=0.95, temperature to 0.9, $repetition\_penalty$ parameter value to 1.35.

4.4 Main Results

Table 1 shows the classification accuracies achieved by DDG in comparison to baseline methodologies across eight commonly used short-text datasets frequently employed in the domain of NLP. The classification accuracies of DDG exhibit varying levels of enhancement relative to the baseline methodologies in the majority of instances. Particularly, datasets such as QQP, MNLI, and QNLI demonstrate a marked performance improvement attributable to the heightened efficiency of DDG in generating more informative textual tokens for simpler text classification datasets characterized by longer textual content. It is important to note that all baseline methodologies are directly derived from the results reported in their respective scholarly publications.

Table 2 presents the classification accuracies comparison between DDG and baseline methodology on two long-text multi-tag datasets, with particular attention to the token length constraint in multi-round ICL settings. Experimental results demonstrate that synthesized samples under equivalent token length limitation consistently outperform baseline methodology across various LLMs, as evidenced by comparative analysis of average accuracy from 1R to 10R shown in Figure 5. Notably, the BANKING77 dataset exhibits textual token lengths ranging 2K-14K (1R-5R) while GoEmotions spans 0.8K-4K for equivalent rounds, with generated demonstrations achieving effective text compression to approximately 70%-75% of original dataset lengths. This compression maintains comparable token lengths between extended round configurations (9R-10R for BANKING77 and 6R-7R for GoEmotions) and the standard 5R benchmark setting. Crucially, our analysis reveals an intrinsic limitation of LLMs in processing redundant inputs: when token lengths exceed a critical threshold, model performance manifests an initial rapid improvement followed by gradual degradation due to parameter overwriting effect. This observation substantiates the methodological necessity of DDG for generating maximally representative demonstrations that balance information density with token efficiency, effectively addressing the forgetting phenomenon while maintaining ICL performance stability across extended round configurations, which is advantageous for LLMs to acquire relevant knowledge from the original dataset for the ICL tasks.

4.5 Ablation Study

Furthermore, we conducted ablation experiments to assess the efficacy of three specific modules within DDG: the implementation of the teacher-student framework (T-S), the utilization of top-k, top-p, and temperature (k/p/T) for the synthesis of samples, and the execution of gradient-descent-based fine-tuning for ϕ parameter (fine-tune). In these ablation experiments, we evaluated each module independently to ascertain

BANKING77	token length					GoEmotions	token length				
LLM	2k	4k	7k	9k	14k	LLM	0.8K	1.6K	2.4K	3.2K	4K
LLaMA-2-7B(DDG)	**36.3**	**72.4**	**77.6**	**81.6**	**86.4**	LLaMA-2-7B(DDG)	**0**	**0.2**	**0.4**	**0.2**	**0.6**
LLaMA-2-7B(LongICLBench)	30.2	70.4	72.0	75.6	77.2	LLaMA-2-7B(LongICLBench)	0	0	0	0.2	0.2
Qwen-1.5-7B(DDG)	**32.8**	**52.6**	**76.2**	**68.4**	**68.0**	Qwen-1.5-7B(DDG)	**15.4**	**18.4**	**19.2**	**19.6**	**15.6**
Qwen-1.5-7B(LongICLBench)	21.6	52.8	61.4	66.0	67.8	Qwen-1.5-7B(LongICLBench)	14.8	18.2	18.6	19.0	14.2
Mistral-7B(DDG)	**37.8**	**66.8**	**70.5**	**71.6**	**74.0**	Mistral-7B(DDG)	**3.6**	**14.4**	**23.6**	**27.0**	**26.8**
Mistral-7B(LongICLBench)	29.8	43.6	66.4	67.8	64.0	Mistral-7B(LongICLBench)	2.6	11.4	7.4	11.6	12.4
Long-LLaMA(DDG)	**4.5**	**24.8**	**38.4**	**42.4**	**36.8**	Long-LLaMA(DDG)	**0**	**0.4**	**1.2**	**1.6**	**2.4**
Long-LLaMA(LongICLBench)	3.0	19.4	28.0	31.6	32.6	Long-LLaMA(LongICLBench)	0	0	0	0.2	0.4

Table 2: Classification accuracies (%) of ICL for DDG and baseline methodology LongICLBench under the same textual token length constraint for BANKING77 and GoEmotions datasets, DDG results are the average of the best three experiments.

T-S	k/p/T	fine-tune	performance of ICL(SST2)
✗	✓	✓	94.0
✓	✓	✗	53.3
✓	✗	✓	92.7
✓	✓	✓	97.3
T-S	k/p/T	fine-tune	performance of ICL(BANKING77)
✗	✓	✓	85.6
✓	✓	✗	42.7
✓	✗	✓	82.0
✓	✓	✓	87.2

Table 3: Classification accuracies (%) of ICL under the ablation experiments setting based on different modules within DDG for SST2 and BANKING77 datasets

the improver model	A_logits	performance of ICL(SST2)
✗	✗	94.0
✓	✗	95.3
✗	✓	94.7
✓	✓	97.3
the improver model	A_logits	performance of ICL(BANKING77)
✗	✗	82.0
✓	✗	86.4
✗	✓	83.6
✓	✓	87.2

Table 4: Classification accuracies (%) of ICL under the ablation experiments setting with different EMA fitting modules for SST2 and BANKING77 datasets

its contribution to the performance enhancement of DDG. We employed the SST2 and BANKING77 datasets, utilizing the LLaMA-2 model under a 5-shot setting for ICL, with classification accuracies serving as the evaluation metric. The results are presented in Table 3.

Based on the statistics from the ablation experiments presented in Table 3, we summarize the following findings: First, the teacher-student framework (T-S) module significantly improves the quality of the synthesized samples by enhancing the stability of the improver model V_{θ_3} trained on these samples. This enhancement facilitates better iterative optimization of the generative model G_ϕ as well. Second, the k/p/T module increases the diversity of synthesized samples while ensuring that these samples contain more valuable information, which is advantageous for optimizing demonstrations contained in the prompt. Lastly, the fine-tune module emerges as the most impactful, as it ensures the grammatical and lexical accuracy of the generated demonstrations, a critical factor for LLMs to effectively acquire relevant knowledge from the input prompt.

We also performed ablation experiments on the Exponential Moving Average (EMA) fitting, which included the improver model V_{θ_3} and the A_logits parameter. The experiments were conducted using the SST2 dataset in conjunction with the LLaMA-2 model, with the demonstrations configured in a 5-shot format. Table 4 illustrates the performance of ICL based on the two EMA modules individually. The results clearly indicate that the incorporation of both the improver model V_{θ_3} and the A_logits parameter contributed to varying degrees of improvement in the classification accuracies of ICL.

In summary, all modules proposed above are contributive to the performance improvement of ICL.

5 Conclusion

This paper proposes a novel Distillation-based Demonstration Generation (DDG) framework, combining with the teacher-student framework, top-k+top-p+temperature approach, which aims to train the generative model to generate distilled synthesized samples that are more representative, and then optimize the prompt and ultimately enhance the performance of ICL. Moreover, the ability to generalize across various LLMs makes DDG valuable for applications on AI research like transfer learning and few-shot learning. Eventually, we designed well-established experiments to validate the superior performance of DDG relative to the state-of-the-art methodologies. The tasks combined in the experiments represent a range of real-world challenges, highlighting the versatility of our approach.

Acknowledgements

Our work was supported in part by the National Natural Science Foundation of China (62202365 and 62132016), Fundamental Research Funds for the Central Universities (QTZX23073), and Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

[Abbasi *et al.*, 2020] Sajjad Abbasi, Mohsen Hajabdollahi, Nader Karimi, and Shadrokh Samavi. Modeling teacher-student techniques in deep neural networks for knowledge distillation. In *MVIP*, 2020.

[Basu *et al.*, 2021] Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *ICLR*, 2021.

[Chen *et al.*, 2023] Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. On the relation between sensitivity and accuracy in in-context learning. In *EMNLP*, 2023.

[Cui *et al.*, 2023] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *ICML*, 2023.

[Dhamala *et al.*, 2023] Jwala Dhamala, Varun Kumar, Rahul Gupta, Kai-Wei Chang, and Aram Galstyan. An analysis of the effects of decoding algorithms on fairness in open-ended language generation. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 655–662, 2023.

[Dong *et al.*, 2024] 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. A survey on in-context learning. In *EMNLP*, 2024.

[Hiraoka *et al.*, 2020] Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Optimizing word segmentation for downstream task. In *EMNLP*, 2020.

[Jiang *et al.*, 2023] Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. Generative calibration for in-context learning. In *EMNLP*, 2023.

[Kaplan *et al.*, 2020] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[Keskar *et al.*, 2019] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[Li and Qiu, 2023] Xiaonan Li and Xipeng Qiu. Finding support examples for in-context learning. In *EMNLP*, 2023.

[Li *et al.*, 2023] Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. In *ACL*, 2023.

[Li *et al.*, 2024] Yichuan Li, Xiyao Ma, Sixing Lu, Kyumin Lee, Xiaohu Liu, and Chenlei Guo. MEND: Meta demonstration distillation for efficient and effective in-context learning. *arXiv preprint arXiv:2403.06914*, 2024.

[Li *et al.*, 2025] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *Transactions on Machine Learning Research*, 2025.

[Liu *et al.*, 2022] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *ACL*, 2022.

[Liu *et al.*, 2024a] Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. se^2: Sequential example selection for in-context learning. In *ACL*, 2024.

[Liu *et al.*, 2024b] Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: making in context learning more effective and controllable through latent space steering. In *ICML*, 2024.

[Liu *et al.*, 2024c] Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*, 2024.

[Lu *et al.*, 2022a] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, 2022.

[Lu *et al.*, 2022b] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *ACL*, 2022.

[Luo *et al.*, 2024] Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y. Zhao. Dr.icl: Demonstration-retrieved in-context learning. *Data Intelligence*, 6(4):909–922, 2024.

[Maekawa *et al.*, 2023] Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. Dataset distillation with attention labels for fine-tuning BERT. In *ACL*, 2023.

[Maekawa *et al.*, 2024] Aru Maekawa, Satoshi Kosugi, Kotaro Funakoshi, and Manabu Okumura. Dilm: Distilling dataset into language model for text-level dataset distillation. In *ACL*, 2024.

[Mao *et al.*, 2023] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *ICML*, 2023.

[Nguyen and Wong, 2023] Tai Nguyen and Eric Wong. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*, 2023.

[Nguyen *et al.*, 2025] Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent llm outputs. In *ICLR*, 2025.

[Peng *et al.*, 2024] Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In *NAACL*, 2024.

[Qin *et al.*, 2024] Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A label is worth a thousand images in dataset distillation. In *NeurIPS*, 2024.

[Robertson and Zaragoza, 2009] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[Rubin *et al.*, 2022] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *NAACL*, 2022.

[S. *et al.*, 2024] Vinay M. S., Minh-Hao Van, and Xintao Wu. In-context learning demonstration selection via influence analysis. *arXiv preprint arXiv:2402.11750*, 2024.

[Scarlatos and Lan, 2024] Alexander Scarlatos and Andrew Lan. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*, 2024.

[Wang *et al.*, 2018] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP*, 2018.

[Wang *et al.*, 2023] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: explaining and finding good demonstrations for in-context learning. In *NeurIPS*, 2023.

[Wu *et al.*, 2023] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *ACL*, 2023.

[Yang *et al.*, 2019] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao. Distillhash: Unsupervised deep hashing by distilling data pairs. In *CVPR*, pages 2946–2955, 2019.

[Yang *et al.*, 2022] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*, 2022.

[Yang *et al.*, 2024] Jinghan Yang, Shuming Ma, and Furu Wei. Auto-icl: In-context learning without human supervision. *arXiv preprint arXiv:2311.09263*, 2024.

[Zhao and Bilen, 2021] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICLR*, 2021.

[Zhao and Bilen, 2023] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *WACV*, 2023.