Demonstration Selection for In-Context Learning via Reinforcement Learning

Xubin Wang [1, 2, 3] Jianfei Wu [3] Yichen Yuan [1] Deyu Cai [1] Mingzhe Li [1] Weijia Jia [1, 3]

[1]Beijing Normal-Hong Kong Baptist University [2]Hong Kong Baptist University [3]Beijing Normal University at Zhuhai

ICML International Conference On Machine Learning

Abstract & Introduction

Large Language Models (LLMs) demonstrate exceptional capabilities in NLP tasks including text annotation, question answering, and dialogue generation. However, enhancing their reasoning capabilities remains crucial for tasks requiring logical reasoning, commonsense understanding, and contextual awareness. In-Context Learning (ICL) provides a promising approach for few-shot learning, enabling LLMs to reason by providing curated demonstrations as context, eliminating extensive retraining requirements.

ICL effectiveness critically depends on selecting appropriate demonstrations from knowledge bases. Conventional methods prioritizing similarity often overlook diversity, leading to biased representations and reduced generalization. These techniques typically employ fixed strategies, failing to dynamically adapt to task-specific requirements.

We introduce the **Relevance-Diversity Enhanced Selection (RDES) framework**, a novel Reinforcement Learning (RL)-based approach optimizing demonstration selection for ICL in few-shot scenarios. RDES leverages RL algorithms to dynamically identify demonstrations maximizing both **diversity** (quantified via label distribution) and **relevance** to task objectives.

Key contributions:

- **RDES framework**: RL-based dynamic demonstration selection enhancing performance and robustness
- **RL optimization**: Balances relevance and diversity to mitigate overfitting
- **CoT integration**: Seamless combination with Chain-of-Thought prompting
- **Comprehensive evaluation**: Outperforms 10 baselines across multiple datasets using 14 LLMs

Figure 1. An example shows how a diversity-based demonstration method works. In this example, the diversity-based method helps the model recognize that the input text expresses a sentiment that is neither strongly positive nor negative, while the no diversity-based method may lead to an inaccurate positive classification due to its lack of varied demonstrations.

Methodology (RL Formulation)

RL provides a natural framework for sequential decision making in demonstration selection. We model the interaction between the selection policy and language model as an iterative process where the policy learns to construct optimal demonstration sets through trial-and-error interactions.

Reinforcement Learning Formulation: We formalize selection as Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

- **State Space ($\mathcal{S}$)**: Captures the complete decision context through four components:
  - Textual features: $\phi_x(x_t) \in \mathbb{R}^{d_x}$ (TF-IDF vector of input text)
  - Demonstration memory: $\phi_E(E_t) \in \mathbb{R}^{d_e}$ (Aggregated embeddings of selected examples)
  - Prediction history: $\phi_y(\hat{y}_t) \in \mathbb{R}^{|\mathcal{Y}|}$ (One-hot encoded previous predictions)
  - Diversity tracking: $D_t = \frac{|\mathcal{L}(E_t)|}{k} \in [1]$ (Normalized label diversity)

  The state embedding is constructed by concatenating these four distinct components:

$$\phi(s_t) = \phi_x(x_t) \oplus \phi_E(E_t) \oplus \phi_y(\hat{y}_t) \oplus \phi_D(D_t) \tag{1}$$

  where $\oplus$ denotes vector concatenation, and each $\phi.$ represents an embedding for the respective component.
- **Action Space ($\mathcal{A}$)**: Discrete selection over candidate demonstrations $\mathcal{K}$, with action $a_t \in \{1, ..., |\mathcal{K}|\}$ indicating the chosen example index from the knowledge base.
- **Transition Dynamics ($\mathcal{P}$)**: Deterministic state updates through demonstration set modification. When action a_t (selecting candidate k_{a_t}) is taken in state $s_t = (x_t, E_t, \hat{y}_t, D_t)$, the next state s_{t+1} becomes:

$$s_{t+1} = f(s_t, a_t) = (x_t, E_t \cup \{k_{a_t}\}, \hat{y}_{t+1}, D_{t+1}) \tag{2}$$

  where $\hat{y}_{t+1}$ is the new prediction based on the updated example set and D_{t+1} is the new diversity score.
- **Reward Function ($\mathcal{R}$)**: A multi-objective reward balancing prediction accuracy and diversity gain:

$$\mathcal{R}(s_t, a_t) = \underbrace{\mathbb{I}(y_{\text{true}} = \hat{y}_t)}_{\text{Accuracy}} + \lambda \underbrace{(D_{t+1} - D_t)}_{\text{Diversity Improvement}} \tag{3}$$

  where $\mathbb{I}(\cdot)$ is the indicator function, y_{true} is the true label, $\hat{y}_t$ is the prediction at step t, D_t is the diversity at step t, D_{t+1} is the diversity after adding the selected example, and λ controls the exploration-exploitation tradeoff. The diversity coefficient λ adapts during training via an annealing schedule:

$$\lambda(t) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min})e^{-\eta t} \tag{4}$$

  This schedule prioritizes early exploration of diverse examples before focusing on accuracy.
- **Discount Factor (γ)**: $\gamma \in [0, 1)$ emphasizes immediate rewards, which is suitable for finite-horizon few-shot learning scenarios where a fixed number of examples are selected.

Methodology (Continued)

RDES advances few-shot learning in NLP by addressing ICL's key challenges through adaptive demonstration selection. By jointly optimizing relevance and diversity, it enhances LLMs' performance on tasks amenable to ICL, particularly classification and reasoning. The schematic framework of RDES is illustrated in Figure 2.

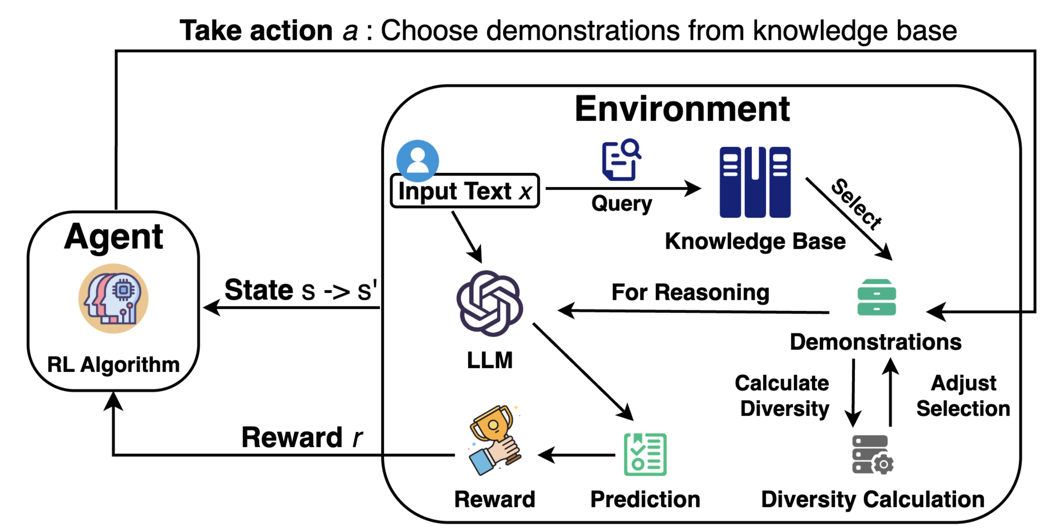

Figure 2. The RDES framework is an adaptive RL approach for few-shot ICL demonstration selection in LLMs. It employs a RL-based agent to dynamically balance the relevance and diversity of selected examples, guided by a reward function that incorporates a label diversity score. This strategy enhances classification accuracy and generalization by mitigating overfitting associated with pure similarity-based methods. The framework involves an Agent interacting with an Environment (including a Knowledge Base and the LLM) to learn an optimal selection policy.

Optimization Framework: Two RL implementations:

- **Q-Learning**:
  - Model-free solution for learning strategies via temporal difference updates.
  - Effective for small/discretizable state spaces.
  - Action-value function $Q(s, a)$ estimates expected cumulative rewards.
  - Updates via standard Q-learning rule: $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$.
  - Implementation aspects: state discretization (TF-IDF binning), ϵ-greedy exploration with exponential decay.
  - Uses tabular Q-value storage.
  - Theoretical convergence under standard conditions (Robbins-Monro, bounded rewards).
- **PPO Variant**:
  - For high-dimensional state spaces where tabular methods are infeasible.
  - Actor-critic architecture using neural networks.
  - **Policy Network (π_θ):** Neural network producing demonstration selection probabilities $\pi_\theta(a|s)$. Uses state embedding $\phi(s)$.
  - **Value Network (V_ψ):** Neural network estimating state value $V_\psi(s)$ (expected cumulative reward). Uses state embedding $\phi(s)$.
  - **Optimization Objective:** PPO optimizes a **clipped surrogate objective** for stability.
  - Combines clipped surrogate loss (L^{CLIP}), value function loss (L^{VF}), and entropy bonus (L^{ENT}): $L(\theta, \psi) = E_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\psi) + c_2 L_t^{ENT}(\theta)]$.
  - L^{CLIP} uses probability ratio $r_t(\theta)$ and advantage estimate A_t, clipped within $[1 - \epsilon, 1 + \epsilon]$.
  - L^{VF} is squared error between predicted value and estimated return $\hat{R}_t$.
  - L^{ENT} encourages exploration.

Algorithm 1 RDES Training Framework

Require: Knowledge base $\mathcal{K}$, Test inputs $\mathcal{D}_{\text{test}}$, LLM $\mathcal{M}$, RL algorithm $\mathcal{A}$
- Initialize selection policy π (Q-table or neural networks)
- Precompute TF-IDF vectors for each sample $x_i \in \mathcal{D}_{\text{test}}$ and for knowledge base $\mathcal{K}$
- **for** $i = 1$ to N **do**
  - Sample input $x_i \in \mathcal{D}_{\text{test}}$
  - Select demonstrations E with initial candidates based on relevance (e.g., top-k TF-IDF matches from $\mathcal{K}$) and apply diversity adjustment
  - Generate prompt $p = \text{Format}(x_i, E)$
  - Obtain prediction $\hat{y} = \mathcal{M}(p)$
  - Compute diversity score $D = \frac{|\mathcal{L}(E)|}{k}$
  - Encode state $s = \phi(x_i, E, \hat{y}, D)$
  - Select action $a \sim \pi(s)$ (example index from $\mathcal{K}$)
  - Calculate reward $r = \mathbb{I}(y_{\text{true}} = \hat{y}) + \lambda(D_{\text{new}} - D_{\text{old}})$
  - Update policy parameters θ using $\mathcal{A}$ with (s, a, r)
- **end for**
- **return** Optimized policy π^*

Prompting Strategies To enhance the performance of the LLM in few-shot settings, we employ two distinct prompting strategies using the selected examples:

- **Standard Prompting:** This strategy constructs a prompt by concatenating the input text, the selected demonstrations (input-output pairs), and the set of possible labels. The LLM is then asked to predict the probability of a label y given this prompt structure:

$$p(y|x, E) = \mathbb{P}_{\text{LM}}\left(y \mid x, \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{k}, \mathcal{Y}\right) \tag{5}$$

  where x is the input text, $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{k}$ are the k selected demonstrations, and $\mathcal{Y}$ is the set of possible labels.
- **CoT Prompting:** This strategy incorporates CoT reasoning into the prompt, allowing the LLM to generate intermediate reasoning steps before producing the final label. This is formulated as marginalizing over possible reasoning chains $\mathcal{R}$:

$$p(y|x, E) = \sum_{r \in \mathcal{R}} \mathbb{P}_{\text{LM}}(r|x, E) \cdot \mathbb{P}_{\text{LM}}(y|x, E, r) \tag{6}$$

  The model first computes the probability of a reasoning chain r given the input and demonstrations, then the probability of the label y conditioned on the input, demonstrations, and the generated reasoning chain.

Experimental Setup

- We evaluated our method using four widely recognized datasets: BANKING77, HWU64, CLINC150, and LIU54, employing a challenge set sampling strategy to ensure rigorous assessment.
- A diverse set of LLMs was utilized, including both closed-source models (e.g., GPT-3.5-turbo) and open-source models (e.g., Gemma-2-2B, LLaMA-3-2-3B), with primary experiments conducted using RDES/B and RDES/C based on the Q-learning framework.
- Additionally, we performed supplementary experiments on challenging benchmarks such as subsets of BigBenchHard, GSM-8K, and SST5, utilizing models known for their strong performance in complex tasks, including Qwen-25-72B and DeepSeek-R1-32B.

Reasoning Performance Analysis

Open-Source Models:

Table 1. Performance comparison of methods designed to boost LLM reasoning across various datasets on open-source LLMs, with a focus on accuracy.

Datasets	Models	Prompt Engineering Methods					Demonstration Selection Methods					Ours	
		ZS	KP	L2M	CoT	SF	FS	FSC	AES	RDS	ADA	RDES/B	RDES/C
BANKING77	Gemma-2-2B	0.200	0.280	0.200	0.200	0.260	0.300	0.340	0.280	0.220	**0.900**	0.831	0.861
	Gemma-2-9B	0.560	0.400	0.500	0.500	0.400	0.440	0.380	0.400	0.400	0.700	0.831	**0.886**
	LLaMA-3.2-1B	0.120	0.100	0.000	0.000	0.100	0.000	0.000	0.020	0.040	0.680	0.024	**0.744**
	LLaMA-3.2-3B	0.200	0.200	0.400	0.500	0.300	0.320	0.060	0.360	0.440	0.700	0.770	**0.805**
	LLaMA-3-8B	0.578	0.560	0.563	0.552	0.458	0.090	0.182	0.531	0.536	0.758	0.784	**0.847**
	Qwen-2.5-7B	0.700	0.480	0.600	0.420	0.480	0.440	0.180	0.440	0.420	0.700	0.803	**0.859**
	Qwen-2.5-14B	0.400	0.400	0.420	0.420	0.420	0.480	0.460	0.500	0.520	0.800	0.839	**0.868**
	Qwen-1.5-72B	0.529	0.480	0.524	0.528	0.551	0.653	0.612	0.509	0.542	0.775	0.785	**0.892**
	Average	0.411	0.363	0.401	0.390	0.371	0.340	0.277	0.380	0.390	0.752	0.708	**0.845**
CLINC150	Gemma-2-2B	0.400	0.600	0.420	0.460	0.560	0.560	0.540	0.500	0.380	0.800	0.875	**0.929**
	Gemma-2-9B	0.700	0.700	0.800	0.800	0.700	0.800	0.680	0.800	0.800	0.780	**0.864**	0.819
	LLaMA-3.2-1B	0.400	0.520	0.060	0.380	**0.600**	0.000	0.020	0.080	0.080	0.400	0.256	0.122
	LLaMA-3.2-3B	0.800	0.700	0.800	0.400	0.700	0.580	0.260	0.600	0.680	0.800	**0.845**	0.703
	LLaMA3-8B	0.523	0.439	0.594	0.504	0.569	0.007	0.285	0.571	0.543	0.767	**0.840**	0.783
	Qwen-2.5-7B	0.740	0.800	0.800	0.780	0.700	0.740	0.460	0.780	0.780	0.800	**0.879**	0.741
	Qwen-2.5-14B	0.900	0.900	0.900	0.800	0.840	0.700	0.700	0.740	0.740	0.900	**0.944**	0.792
	Qwen-1.5-72B	0.726	0.517	0.683	0.641	0.660	0.850	0.652	0.696	0.656	0.861	0.897	**0.963**
	Average	0.649	0.647	0.632	0.596	0.666	0.530	0.450	0.596	0.582	0.763	**0.800**	0.731
HWU64	Gemma2-2B	0.300	0.320	0.300	0.300	0.400	0.460	0.440	0.420	0.360	0.600	0.832	**0.851**
	Gemma2-9B	0.600	0.600	0.600	0.600	0.600	0.700	0.700	0.700	0.700	0.800	0.877	**0.910**
	LLaMA-3.2-1B	0.200	0.100	0.080	0.000	0.100	0.020	0.000	0.020	0.060	0.360	0.381	**0.687**
	LLaMA-3.2-3B	0.300	0.100	0.300	0.200	0.300	0.220	0.180	0.300	0.300	0.700	0.747	**0.817**
	LLaMA-3-8B	0.478	0.407	0.493	0.479	0.563	0.632	0.498	0.651	0.645	0.837	0.816	**0.859**
	Qwen-2.5-7B	0.780	0.700	0.800	0.600	0.800	0.640	0.540	0.760	0.740	0.800	0.805	**0.880**
	Qwen-2.5-14B	0.780	0.800	0.800	0.440	0.740	0.800	0.800	0.720	0.680	**0.900**	0.886	0.895
	Qwen-1.5-72B	0.698	0.615	0.676	0.661	0.668	0.825	0.817	0.749	0.774	0.877	0.867	**0.924**
	Average	0.517	0.455	0.506	0.410	0.521	0.537	0.497	0.540	0.532	0.734	0.776	**0.853**
LIU54	Gemma-2-2B	0.400	0.400	0.500	0.400	0.400	0.620	0.480	0.500	0.440	0.600	0.733	**0.854**
	Gemma-2-9B	0.500	0.500	0.600	0.600	0.600	0.580	0.580	0.500	0.500	**1.000**	0.722	0.837
	LLaMA-3.2-1B	0.200	0.160	0.300	0.400	0.080	0.040	0.040	0.360	0.320	**0.700**	0.058	0.651
	LLaMA-3.2-3B	0.400	0.400	0.400	0.300	0.400	0.360	0.320	0.500	0.400	0.600	**0.772**	0.749
	LLaMA-3-8B	0.358	0.409	0.428	0.360	0.392	0.396	0.320	0.347	0.312	0.763	0.779	**0.811**
	Qwen-2.5-7B	**0.800**	0.700	0.700	0.640	0.520	0.620	0.500	0.500	0.660	**0.800**	0.794	0.765
	Qwen-2.5-14B	0.700	0.860	0.700	0.660	0.700	0.660	0.600	0.740	0.780	**1.000**	0.849	0.743
	Qwen-1.5-72B	0.496	0.445	0.487	0.491	0.550	0.609	0.647	0.514	0.492	0.769	0.781	**0.880**
	Average	0.482	0.484	0.514	0.481	0.455	0.486	0.436	0.495	0.488	0.779	0.686	**0.786**

Closed-Source Models:

Table 2. Performance comparison of methods designed to boost LLM reasoning across various datasets on closed-source LLMs, with a focus on accuracy.

Datasets	Models	Prompt Engineering Methods					Demonstration Selection Methods					Ours	
		ZS	KP	L2M	CoT	SF	FS	FSC	AES	RDS	ADA	RDES/B	RDES/C
BANKING77	GPT-3.5-turbo	0.340	0.240	0.260	0.200	0.380	0.520	0.320	0.260	0.240	0.360	0.767	**0.858**
	Doubao-lite-4k	0.300	0.300	0.300	0.320	0.300	0.500	0.360	0.300	0.280	0.400	0.750	**0.830**
	Doubao-pro-4k	0.500	0.400	0.500	0.480	0.600	0.540	0.540	0.700	0.680	**0.900**	0.838	0.888
	Hunyuan-lite	0.300	0.233	0.433	0.200	0.300	0.233	0.133	0.320	0.320	0.600	0.593	**0.775**
	Average	0.360	0.293	0.373	0.300	0.395	0.448	0.338	0.395	0.380	0.565	0.737	**0.838**
CLINC150	GPT-3.5-turbo	0.460	0.420	0.400	0.480	0.460	0.600	0.380	0.300	0.380	0.720	0.845	**0.949**
	Doubao-lite-4k	0.700	0.600	0.600	0.700	0.500	0.680	0.440	0.680	0.660	0.700	0.825	**0.927**
	Doubao-pro-4k	0.660	0.680	0.620	0.700	0.700	0.800	0.640	0.680	0.640	0.900	0.938	**0.961**
	Hunyuan-lite	0.633	**0.800**	0.767	0.700	0.633	0.467	0.500	0.480	0.620	**0.800**	0.730	0.772
	Average	0.613	0.625	0.597	0.645	0.573	0.637	0.490	0.535	0.575	0.780	0.835	**0.902**
HWU64	GPT-3.5-turbo	0.260	0.360	0.280	0.340	0.280	0.560	0.360	0.100	0.260	0.520	0.850	**0.914**
	Doubao-lite-4k	0.500	0.500	0.500	0.480	0.500	0.520	0.340	0.360	0.420	0.700	0.765	**0.873**
	Doubao-pro-4k	0.640	0.760	0.620	0.800	0.640	0.680	0.600	0.620	0.640	**1.000**	0.862	0.918
	Hunyuan-lite	0.533	0.367	0.333	0.433	0.233	0.600	0.433	0.540	0.320	0.700	0.514	**0.784**
	Average	0.483	0.497	0.433	0.513	0.413	0.590	0.433	0.405	0.410	0.730	0.748	**0.872**
LIU54	GPT-3.5-turbo	0.380	0.260	0.360	0.460	0.240	0.480	0.480	0.140	0.180	0.300	0.743	**0.868**
	Doubao-lite-4k	0.500	0.400	0.500	0.540	0.660	0.600	0.440	0.520	0.520	0.600	0.690	**0.841**
	Doubao-pro-4k	0.400	0.420	0.400	0.520	0.520	0.800	0.760	0.500	0.520	**0.900**	0.829	0.884
	Hunyuan-lite	0.533	0.500	0.567	0.700	0.633	0.367	0.500	0.460	0.620	0.560	0.565	**0.704**
	Average	0.453	0.395	0.457	0.555	0.513	0.562	0.545	0.405	0.460	0.590	0.707	**0.824**

- This section provides a comprehensive evaluation of the reasoning accuracy of both closed-source and open-source LLMs across four benchmark datasets, utilizing various prompt engineering and demonstration selection techniques. The evaluation highlights popular closed-source models such as GPT-3.5-turbo and Doubao, alongside open-source alternatives like Gemma, LLaMA, and Qwen, with results presented in tables that emphasize the top-performing techniques in **bold** and the second-best results as underlined for clarity.
- **Prompt Engineering Methods:**
  - Zero-Shot (ZS): No demonstrations, tests generalization.
  - Knowledge Prompting (KP): Contextual information generation.
  - Least-to-Most (L2M): Break tasks into smaller steps.
  - Chain of Thought (CoT): Step-by-step reasoning.
  - Self-Refine (SF): Iterative critique and refinement.
- **Demonstration Selection Methods:**
  - Few-Shot (FS): Limited text-label pairs.
  - Few-Shot with CoT (FSC): FS + explanations.
  - Active Selection (AES): Iterative selection using RL.
  - Representative Selection (RDS): Diverse subset identification.
  - Adaptive Selection (ADA): Uncertainty-based, semantic diversity.
- **Ours:** RDES/B (base), RDES/C (RDES/B + CoT), RDES/PPO (PPO variant).

Figure 3. **Average performance**: The data summarizes the average performance results of various LLMs, encompassing both closed-source and open-source variants, across different datasets. RDES variants outperform prompt engineering (PE) and demonstration selection (DS) baselines.

Conclusion

- We introduced RDES, a novel framework that employs reinforcement learning (specifically Q-learning and a PPO-based variant) to optimize demonstration selection for in-context learning in LLMs by balancing relevance and diversity, enhancing generalization and mitigating overfitting.
- Our extensive evaluation against ten baselines on four benchmark classification datasets demonstrated that RDES significantly outperforms existing methods, with integration of RDES and CoT reasoning (RDES/C) generally improving performance, though benefits vary by model and dataset.
- Additional experiments on challenging reasoning benchmarks and varying demonstration counts further validated RDES's effectiveness, particularly the RDES/PPO variant, highlighting its potential for adaptive demonstration selection in complex NLP tasks.
- Future work will focus on refining diversity metrics, extending RDES to other tasks like generation and question answering, making CoT usage adaptive, analyzing computational efficiency, exploring different retrieval methods, and assessing generalization across datasets.

Acknowledgements

Supported by: NSFC (62272050, 62302048), Guangdong Key Lab of AI, Zhuhai Sci-Tech Innovation Bureau (2320004002772), and Beijing Normal University (Zhuhai) Computing Resources.

 wangxubin@ieee.org