Reward Mixology: Crafting Hybrid Signals for Reinforcement Learning Driven In-Context Learning

Changshuo Zhang[1]* **Ang Gao**[1]* **Xiao Zhang**[1]† **Yong Liu**[1]
Deyang Li[2] **Fangchao Liu**[2] **Xinyu Zhang**[2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Huawei Poisson Lab, China

Abstract

In-context learning (ICL) performance heavily relies on the quality and ordering of demonstrations. Iterative selection (IS) is a promising approach to address this issue, but existing IS methods face two key challenges: the oversimplification of process reward signals that guide intermediate steps (often using single-dimensional metrics) and the lack of outcome reward signals that directly optimize final-task accuracy (relying solely on binary terminal feedback like correct/incorrect predictions). To address these issues, we propose a reinforcement learning method R-Mix which models iterative demonstration selection as a Markov Decision Process (MDP), crafting hybrid reward signals — combining outcome-based accuracy signals (i.e., outcome rewards) with process-oriented signals (i.e, process rewards) like step-wise influence and label entropy improvement. Our analysis reveals a positive but trade-off relationship between outcome rewards and process rewards, underscoring the importance of both components for effective policy optimization. We further introduce a dual-head policy architecture that explicitly decouples input-semantic relevance and label-content compatibility. Experiments across NLP benchmarks demonstrate superior performance over state-of-the-art methods, with ablation studies validating the necessity of both reward components and architectural disentanglement. Our work has deeply explored the effective potential of ICL through demonstration selection.

1 Introduction

In-context learning (ICL), an emergent ability of large language models (Brown et al., 2020), enables these models to achieve impressive performance on diverse tasks—such as text summarization, dialogue management, and semantic analysis—by merely incorporating demonstration input–label pairs as prompts, all without any parameter updates (Sia and Duh, 2023; Li et al., 2023; He et al., 2023; Sun et al., 2024; Liu et al., 2024b; Chen et al., 2023; Shen et al., 2024).

Demonstration selection critically impacts ICL efficacy, where demonstrations can be manually designed or retrieved from datasets. Research has demonstrated that both the quality and order of demonstrations significantly influence the ICL capabilities of large language models (Gao et al., 2020; Lu et al., 2021). While point-wise similarity matching (Liu et al., 2021; Reimers and Gurevych, 2019) efficiently retrieves test-relevant demonstrations, it neglects inter-demonstration synergies. Sample-then-select methods (Guo et al., 2024; Lu et al., 2021; Wu et al., 2022) address this via permutation sampling and list-wise metrics (e.g., permutation entropy, mutual information), yet suffer from computational overhead and sampling bias. Recent iterative approaches (Liu et al., 2024a; Chen et al., 2024; Liu et al., 2024b), while circumventing these issues, adopt a myopic optimization strategy—exclusively relying on oversimplified intermediate benefits (referred to as process rewards)—that fails to align with the improvement of final-task performance. This creates a critical gap that hinders unleashing the effective potential of LLMs for ICL through demonstration selection.

In response to these challenges, we model the demonstration selection process as a Markov Decision Process (MDP) and optimize it by training a small reinforcement learning model (named R-Mix) utilizes crafted hybrid reward signals. These include an outcome reward (OR) aligned with final-task accuracy, which however suffers from sparsity since it is only obtainable after the entire selection process is completed. To mitigate optimization difficulties associated with sparse rewards ((Hare, 2019; Devidze et al., 2022)), we introduce several rich process rewards (PR), such as improvements in influence and label entropy. We analyze the re-

*Equal contribution.
†Corresponding author (e-mail: zhangx89@ruc.edu.cn).

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 4373–4383
November 4-9, 2025

lationship between process rewards and outcome rewards, finding that while they exhibit a positive correlation, it's also a trade-off, highlighting the need for all types of rewards to guide the policy effectively. Furthermore, since both the input and output in the demonstrations affect the final selection, we adopt a dual-head policy model to decouple their respective contributions. We validate the effectiveness of our model on three datasets and provide extensive experimental analysis to demonstrate its robustness and generalizability.

2 Related Work

Demonstration selection can critically affect the performance of ICL. Several studies have proposed methods that leverage point-wise matching to retrieve demonstrations that are most relevant to the task at hand (Liu et al., 2021; Reimers and Gurevych, 2019; Rubin et al., 2021). Uncertainty (Diao et al., 2023; Hübotter et al., 2024) , influences (Nguyen and Wong, 2023), entropy (Lu et al., 2021; Wu et al., 2022), diversity (Ye et al., 2023b; Su et al., 2023), sensitivity (Chen et al., 2022) and LLM scores (Ye et al., 2023a; Zhang et al., 2023) have also been utilized as criteria for demostration selection. While these methods are efficient, they fail to account for the interactions and synergies between different demonstrations.

Order sensitivity plays a crucial role in maximizing ICL performance. Research has shown that the sequence in which demonstrations are presented significantly influences model behavior (Gao et al., 2020; Lu et al., 2021). Ordering strategies based on list-wise metrics, such as permutation entropy and mutual information, help capture the relationships between demonstrations, optimizing the presentation for the model (Guo et al., 2024; Wu et al., 2022). However, these sample-then-select methods often suffer from computational overhead and potential biases introduced by the sampling process, raising concerns about their scalability and efficiency in practical applications.

Iterative methods have been proposed to optimize demonstration selection and ordering in a more dynamic and adaptive manner. These methods focus on refining the demonstration sequence iteratively (Liu et al., 2024a; Zhang et al., 2022a; Chen et al., 2024; Liu et al., 2024b). Liu et al. (2024a) trains a ranker to iteratively approximate the optimal demonstration list. Reinforcement Learning (RL) has also been explored as a key technique for optimizing demonstration selection through iterative refinement with process influence improvement (Zhang et al., 2022a; Chen et al., 2024). While such iterative approaches have shown promise in improving performance, they are optimized using intermediate benefits (referred to as process rewards) and lack outcome-oriented reward signals to directly optimize final-task performance. Additionally, their process rewards remain oversimplified. Our work builds on this gap by introducing outcome rewards with process rewards that capture intermediate activation and guide the model to refine demonstration selection.

3 Reward MIXolog (R-Mix): Our Method

To address the challenge of example selection in ICL, our K-shot iterative demonstration retrieval method models the demonstration selection process as a Markov Decision Process and optimizes the policy based on the hybrid reward, as shown in Figure 1. RMix operates through three components: (1) an *MDP formulation* modeling demonstration selection (Sec. 3.1), (2) a *dual-head scoring policy* that disentangles the contribution of input and label semantics respectively, and (3) a *hybrid reward system* combining outcome accuracy gains with stepwise influence and entropy signals whose positive but trade-off correlation is analyzed in Figs. 2(a)-2(b). The reward shaping via batch normalization (Sec. 3.4) ensures stable policy optimization.

3.1 Iterative Demonstration Retrieval Formulation

Given a pretrained language model LM, a retrieval model R, a candidate demonstration set $D = \{(x_i, y_i)\}_{i=1}^{N}$, and a query input x_{test}, the goal is to iteratively construct an ordered demonstration sequence $S_K = [(x_{s_1}, y_{s_1}), \ldots, (x_{s_K}, y_{s_K})]$ that maximizes the K-shot ICL performance gain:

$$S_K^* = \underset{S_K \subseteq D,\, |S|=K}{\arg\max} \Big[\mathbb{I}\left(\text{LM}\left(S_K \oplus x_{\text{test}}\right) = y_{\text{true}}\right) - \mathbb{I}\left(\text{LM}\left(x_{\text{test}}\right) = y_{\text{true}}\right) \Big], \tag{1}$$

where $S \oplus x_{\text{test}}$ represents the concatenation of the demonstration examples S with the test input x_{test}, and y_{true} indicates the ground-truth label for the given demonstration set.

MDP Formulation This iterative process can be formally modeled as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$:

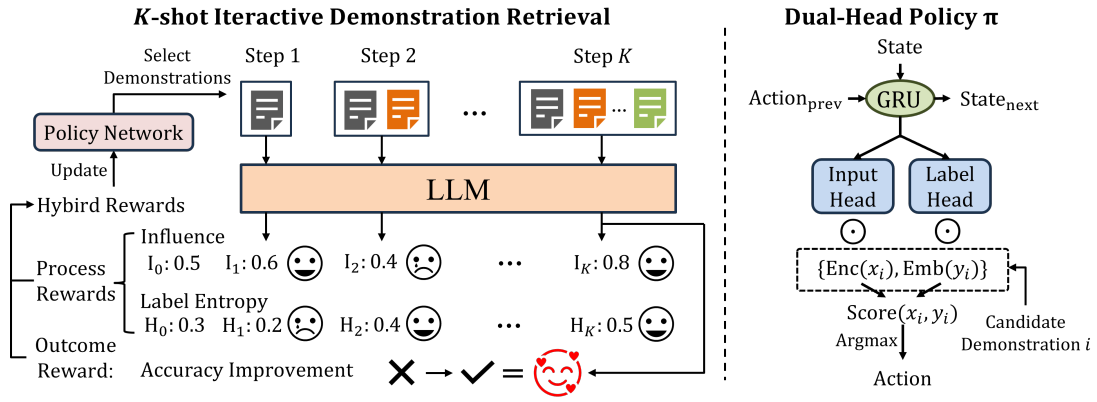


Figure 1: Our proposed R-Mix: A K-shot iterative demonstration retrieval method employing a dual-head policy. The architecture dynamically selects demonstrations through iterative scoring K times, where the input head computes content scores by evaluating BERT-encoded demonstration inputs $\text{Enc}(x_i)$ from the candidate set, while the label head calculates label scores through learnable embeddings $\text{Emb}(y_i)$. Their fused compatibility scores (Eq. (9)) govern the policy's selection probabilities, optimized by hybrid rewards (outcome reward for terminal accuracy and process reward for stepwise signals).

- **State Space** $\mathcal{S}$: State $s_t \in \mathcal{S}$ at step t consists of:
  - Current demonstration sequence $S_{t-1} = [(x_{s_1}, y_{s_1}), \ldots, (x_{s_{t-1}}, y_{s_{t-1}})]$.
  - Query input x_{test}.
  - GRU internal state $\mathbf{s}_x^t$ (from Sec. 3.2)

  Formally: $s_t = (S_{t-1}, x_{\text{test}}, \mathbf{s}_x^t)$.

- **Action Space** $\mathcal{A}$: At step t, select from remaining candidates:

$$\mathcal{A}_t = D \setminus S_{t-1}, \quad a_t = (x_i, y_i) \in \mathcal{A}_t. \tag{2}$$

- **Transition Dynamics** $\mathcal{P}$: Deterministic state update through concatenation:

$$\mathcal{P}(s_{t+1}|s_t, a_t) = \begin{cases} 1 & \text{if } S_t = S_{t-1} \oplus (x_i, y_i), \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

  where $\oplus$ denotes sequence concatenation.

- **Reward Function** $\mathcal{R}$: Combines immediate and terminal rewards:

$$\mathcal{R}(s_t, a_t) = \begin{cases} \lambda \mathcal{R}_{\text{pr}}^t & t < K, \\ \mathcal{R}_{\text{or}}(S_K) + \lambda \mathcal{R}_{\text{pr}}^{(K)} & t = K, \end{cases} \tag{4}$$

  where $\mathcal{R}_{\text{pr}}^t$ and $\mathcal{R}_{\text{or}}$ are designed follows Sec.3.3.

- **Discount Factor** γ: Exponential reward discounting with $\gamma \in [0, 1]$ as in Eq. (16).

Policy Implementation The dual-head architecture generates selection probabilities:

$$\pi_\theta(a_t = (x_i, y_i)|s_t) = \frac{\exp(\text{Score}_R^t(x_i, y_i))}{\sum_{(x_j, y_j) \in \mathcal{A}_t} \exp(\text{Score}_R^t(x_j, y_j))}, \tag{5}$$

where Score_R^t combines input and label scores through Eq. (9).

Construction Process The iterative demonstration retrieval process is formalized in Algorithm 1. This formulation enables end-to-end optimization through reinforcement learning.

3.2 Dual-Head Policy Architecture

The dual-head policy network integrates historical information through a hybrid embedding mechanism. Specifically, it first encodes the previous input x^{t-1} using BERT and then concatenates this with the embedding of the previous label y^{t-1}, which is obtained via a learnable embedding matrix $\mathbf{E}_{\text{label}} \in \mathbb{R}^{|\mathcal{Y}| \times d_y}$. This concatenated vector,

$$\mathbf{h}^{t-1} = \text{BERT}(x^{t-1}) \oplus \mathbf{E}_{\text{label}}(y^{t-1}), \tag{6}$$

has a dimension $d_h = d_x + d_y$, where d_x is the dimensionality of the BERT-encoded input and d_y is that of the label embedding. The hybrid embedding $\mathbf{h}^{t-1}$ is then fed into a Gated Recurrent Unit (GRU) cell together with the previous hidden state $\mathbf{s}_x^{t-1}$ to produce the output $\mathbf{o}^t$ and update the state to $\mathbf{s}_x^t$:

$$\mathbf{o}^t, \mathbf{s}_x^t = \text{GRU}_\phi\left(\mathbf{h}^{t-1}, \mathbf{s}_x^{t-1}\right). \tag{7}$$

Algorithm 1 Iterative Demonstration Retrieval

Require: Pretrained LM M, retrieval model R, candidate set $D = \{(x_i, y_i)\}_{i=1}^{N}$, query x_{test}, maximum sequence length K

Ensure: Demonstration sequence $S = [(x_{s_1}, y_{s_1}), \ldots, (x_{s_K}, y_{s_K})]$

Initialize $S_0 \leftarrow \emptyset$, $t \leftarrow 1$
while $t \leq K$ **do**
  Compute relevance scores:
  for each $(x_i, y_i) \in D \setminus S_{t-1}$ **do**
    $\text{Score}_R^t(x_i, y_i) \leftarrow R\big((x_i, y_i), x, \mathbf{s}_x^t\big)$
  end for
  Demonstration Selection:
  if training **then** ▷ Stochastic Sampling
    $(x_{s_t}, y_{s_t}) \leftarrow \text{Sample}(\text{Score}_R^t(x_i, y_i))$
  else ▷ Greedy Selection
    $(x_{s_t}, y_{s_t}) \leftarrow \text{Argmax}(\text{Score}_R^t(x_i, y_i))$
  end if
  Update sequence:
  $S_t \leftarrow S_{t-1} \oplus (x_{s_t}, y_{s_t})$
  $t \leftarrow t + 1$
end while
return S_K

At the start, the previous input is initialized as x and the label embedding is set to a zero vector.

Then we compute compatibility scores through dual projection heads:

$$\begin{aligned} \text{Score}_x^t(x_i) &= \langle \text{BERT}(x_i), \mathbf{W}_x \mathbf{o}^t \rangle / \sqrt{d_x}, \\ \text{Score}_y^t(y_i) &= \langle \mathbf{E}_{\text{label}}(y_i), \mathbf{W}_y \mathbf{o}^t \rangle / \sqrt{d_y}, \end{aligned} \tag{8}$$

with learnable projection matrices $\mathbf{W}_x \in \mathbb{R}^{d_x \times d_h}$, $\mathbf{W}_y \in \mathbb{R}^{d_y \times d_h}$. The $\sqrt{d_x}$ and $\sqrt{d_y}$ term stabilizes gradient magnitudes.

For fusion $\text{Score}_x^t(x_i)$ and $\text{Score}_y^t(y_i)$ adaptively, we simply add them together:

$$\text{Score}_R^t(x_i, y_i) = \text{Score}_x^t(x_i) + \text{Score}_y^t(y_i). \tag{9}$$

To select the action a_t at step t, our model uses the composite score $\text{Score}_R^t(x_i, y_i)$ defined in Eq. (9). During training, we employ Thompson sampling (Russo et al., 2018) for action selection to ensure proper exploration (Ladosz et al., 2022; Zhang et al., 2024; Chen et al., 2023), while at inference time, we simply take the greedy action by selecting the maximum-scoring demonstration pair. This balanced approach maintains exploration during learning while guaranteeing optimal performance during deployment.

3.3 Hybrid Reward Design

To guide the iterative construction of demonstration examples, we propose a dual-component reward system that synergistically combines global optimization with stepwise guidance:

Outcome Reward (Global Optimization) This component quantifies the terminal performance improvement induced by the constructed demonstration S:

$$\begin{aligned} \mathcal{R}_{\text{or}}(S_K) = \mathbb{I}\left(\text{LM}\left(S_K \oplus x_{\text{test}}\right) = y_{\text{true}}\right) \\ - \mathbb{I}\left(\text{LM}\left(x_{\text{test}}\right) = y_{\text{true}}\right), \end{aligned} \tag{10}$$

where $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition is satisfied and 0 otherwise, and $S \oplus x$ denotes the concatenation of demonstration S with test input x.

Process Reward (Stepwise Guidance) R-Mix faces a sparsity issue with its outcome-dependent reward mechanism, which only provides binary signals (0/+1/-1) upon task completion, failing to guide intermediate optimization processes and leading to susceptibility to local optima (Hare, 2019; Devidze et al., 2022). To address this, we propose introducing process rewards that decompose the task into learnable incremental steps, thereby providing continuous learning signals throughout the construction phase. Specifically, the process reward will be restructured into two elements:

- **Influence Improvement**: Measures the local performance gain at step t, which was introduced in (Zhang et al., 2022b) to quantify the improvement in the ground truth probability for the test example brought by the selected sample at each step:

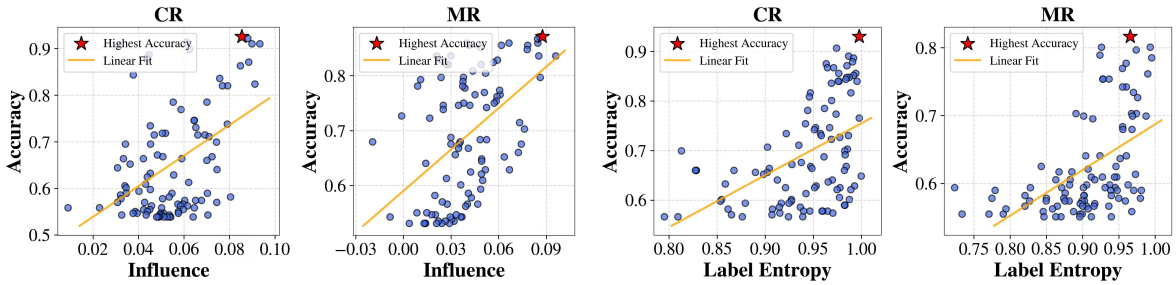
$$\mathcal{R}_{\text{infl}}^t = I(S_t) - I(S_{t-1}), \tag{11}$$

where

$$I(S) = P_{\text{LM}}(y_{\text{true}} | S \oplus x_{\text{test}}). \tag{12}$$

- **Label Entropy Improvement**: Recent studies suggest that effective example selection should encourage the language model's output distribution for empty inputs to approach uniformity (Lu et al., 2021; Guo et al., 2024). We quantify uncertainty reduction through entropy differentials:

$$\mathcal{R}_{\text{entr}}^t = H\left(S_t\right) - H\left(S_{t-1}\right), \tag{13}$$



(a) Relation between influence and accuracy.

(b) Relation between label entropy and accuracy.

Figure 2: Relation between process rewards (influence and label entropy) with outcome rewards (accuracy).

where $H(\cdot)$ denotes the Shannon entropy of the model's output distribution over the label space $\mathcal{Y}$ when provided with an empty query "N/A":

$$\begin{aligned} H(S) &= E\left(P_{\text{LM}}(\cdot|S \oplus \text{“N/A”})\right) \\ &= -\sum_{y \in \mathcal{Y}} P(y) \log P(y). \end{aligned} \tag{14}$$

The composite process reward at each step aggregates these components:

$$\mathcal{R}_{\text{pr}}^t = \mathcal{R}_{\text{infl}}^t + \mathcal{R}_{\text{entr}}^t. \tag{15}$$

Interplay Between Rewards As shown in Figure 2(a) and Figure 2(b), process rewards exhibit a positive but trade-off relationship with the outcome reward across both CR (Hu and Liu, 2004) and MR (Pang and Lee, 2005) datasets. Specifically:

- **Positive Correlation**: In the initial phase, higher cumulative influence improvement ($\sum \mathcal{R}_{\text{infl}}^t$) and label entropy ($\sum \mathcal{R}_{\text{entr}}^t$) generally correspond to improved final accuracy ($\mathcal{R}_{\text{or}}$). For example, in the CR dataset, trajectories with $\sum \mathcal{R}_{\text{infl}}^t > 0.02$ achieve $\mathcal{R}_{\text{or}} \geq 0.55$, while those with $\sum \mathcal{R}_{\text{infl}}^t < 0$ yield $\mathcal{R}_{\text{or}} \leq 0.51$.
- **Trade-off Relationship**: Beyond sufficiently high process rewards, further increases may lead to reduced accuracy as shown in Figure 2(a): when Influence exceeds 0.08, accuracy drops significantly. This likely occurs because the LLM allocates greater influence to examples that were already correctly answered initially. As illustrated in Figure 2(b), this phenomenon becomes more pronounced—blindly emphasizing label entropy does not reliably improve accuracy, underscoring the need for balanced reward design that integrates procedural guidance with outcome optimization.

These phenomenons highlight the need for a hybrid approach: outcome rewards provide a global optimization target, while process rewards offer detailed guidance for each step. Together, they address the challenges of sparse rewards and align the policy learning process with the dynamic reasoning patterns of LLMs.

3.4 Policy Optimization

We optimize the policy network using a stabilized REINFORCE algorithm with two key enhancements: temporal discounting and in-batch reward normalization. The objective function maximizes the expected discounted cumulative reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{S \sim \pi_\theta}\left[\sum_{t=1}^{K} \gamma^{t-1} \mathcal{R}^t\right], \tag{16}$$

where $\gamma \in [0, 1]$ is the discount factor, and the step reward $\mathcal{R}^t$ combines immediate process reward signals and terminal outcome reward signals:

$$\mathcal{R}^t = \mathbb{I}(t = K) \cdot \underbrace{\mathcal{R}_{\text{or}}(S_K)}_{\text{Outcome Reward}} + \underbrace{\lambda \cdot \mathcal{R}_{\text{pr}}^t}_{\text{Process Reward}}, \tag{17}$$

where λ controls their relative importance. To stabilize training dynamics, we design the advantage function to combine long-term outcome rewards with discounted process rewards:

$$\mathcal{A}^t = \gamma^{K-t} \mathcal{R}_{\text{or}}(S) + \lambda \cdot \sum_{i=t}^{K} \gamma^{i-t} \mathcal{R}_{\text{pr}}^{(i)}. \tag{18}$$

To mitigate the non-stationarity of reward distributions across batches and significantly enhance learning efficiency (Naik et al., 2024; Hu, 2025), we apply in-batch standardization:

$$\hat{\mathcal{A}}^t = \frac{\mathcal{A}^t - \mu_{\mathcal{A}}}{\sigma_{\mathcal{A}}}, \tag{19}$$

Method	CR	MR	SST2
Zero-shot	0.5586	0.6211	0.6328
Random	0.5625	0.5703	0.5547
Similarity	0.6250	0.6594	0.6872
Best-of-N	0.5508	0.5391	0.7695
MDL	0.5352	0.5547	0.6352
MI	0.5039	0.7930	0.7891
GLE	0.7659	0.7891	0.7820
DEmO	0.8984	0.7789	0.8711
EPR	0.6477	0.7143	0.7477
LLM-R	0.6831	0.6460	0.7246
Se^2	0.8553	0.7443	0.8253
IterR	0.8828	0.8398	0.8789
R-Mix	**0.9062**	**0.9023**	**0.9102**

Table 1: 8-shots performance on various methods.

where $\mu_{\mathcal{A}}$ and $\sigma_{\mathcal{A}}$ denote the batch-wise mean and standard deviation of advantage values. This normalization scheme ensures gradient estimates remain scale-invariant across different task domains. The final policy gradient is computed as:

$$\nabla_\theta \mathcal{J}(\theta) \approx \frac{1}{B} \sum_{b=1}^{B} \sum_{t=1}^{K} \hat{\mathcal{A}}_b^t \nabla_\theta \log \pi_\theta(a_b^t | S_b^{t-1}, x_b), \tag{20}$$

where B is the batch size and K is the maximum demonstration length.

4 Experiments

4.1 Setup

Building upon previous work (Guo et al., 2024), we adopt the LLaMA2 series models (Xia et al., 2023; Touvron et al., 2023) as the inference LLM in our study. The majority of the analyses are conducted using Sheared LLaMA2 1.3B. We evaluate the model on three textual classification task datasets and compare its performance with several existing methods. The specific details of the experimental setup are outlined as follows.

Datasets: Given previous research (Guo et al., 2024; Yang et al., 2023; Wu et al., 2022; Lu et al., 2021), we conducted experiments on three datasets, covering CR (Hu and Liu, 2004), MR (Pang and Lee, 2005) and SST2 (Socher et al., 2013).

Compared Methods: We selected several baselines as comparison approaches, including both *learning-free* and *learning-based* methods, *point-wise* and *list-wise* methods. For *learning-free* methods, we selected:

- **Zero-shot** method directly applies a LLM to perform classification tasks without additional guidance. **Random** serves as a baseline by randomly selecting and ordering the dataset.
- The *point-wise* **Similarity** method (Liu et al., 2022) selects demonstrations based on their semantic similarity to the current query.
- *List-wise* methods: **Best-of-N** generates N random dataset permutations and selects the highest-accuracy configuration. The **MDL** approach (Wu et al., 2022) creates permutations per instance and chooses the one minimizing label encoding codelength. The **MI** method (Sorensen et al., 2022) uses information theory to optimize demonstration selection, while **GLE** (Lu et al., 2021) reduces order sensitivity via entropy-based metrics. Furthermore, **DEmO** (Guo et al., 2024) determines optimal demonstration order without external data using dataset-free optimization.

For *learning-based* methods, we selected:

- *Point-wise* methods: **EPR** (Rubin et al., 2021) uses a scoring LM to label examples and trains a contrastive dense retriever. **LLM-R** (Wang et al., 2023) trains a reward model for scoring, then distills knowledge into a retriever.
- *List-wise* methods: **Se^2** (Liu et al., 2024a) trains a ranker to iteratively approximate the optimal demonstration list. Lastly, **IterR** (Chen et al., 2024) iteratively refines the demonstration selection process using a reinforcement learning-based retriever.

Evaluations: For each dataset, we randomly selected a subset of 256 samples from the validation set as the test set. We use the LLM accuracy on the test set as the evaluation metric.

Implementation Details: We randomly construct a demonstration pool consisting of 512 samples, from which we select 8 instances iteratively to formulate an 8-shot learning. For point-wise methods, we sort demonstrations by scores in descending order. For sample-then-select methods, we set the sample number to 100. For learning-based methods, we set the learning rate

to $1e^{-3}$ for 100 epochs, ensuring stable convergence and effective policy optimization. λ is chosen from $\{0.1, 0.5, 1.0, 2.0\}$. γ is chosen from $\{0.8, 0.9, 0.95\}$.

4.2 Main Results

Our experimental results reveal four critical insights about in-context demonstration selection:

(1) The Random baseline achieves lower accuracy than Zero-shot on MR (56.25% vs. 62.11%) and SST2 (55.47% vs. 63.28%), indicating that arbitrary demonstration ordering can degrade LLM performance. In contrast, Similarity-based selection shows consistent improvements (e.g., +6.64% on CR), suggesting targeted demonstration curation is crucial for effective in-context learning.

(2) Sample-then-select methods demonstrate significant gains over Zero-shot, with DEmO achieving second-best performance through list-wise metrics: 89.84% on CR and 87.11% on SST2. This validates the importance of holistic demonstration list evaluation beyond individual sample quality.

(3) Learning-based methods like EPR and LLM-R outperform most learning-free approaches by adapting to task-specific data, but they show overall inferiority compared to *list-wise* methods due to neglecting interactions between examples.

(4) Se^2 demonstrates high performance, yet its greedy-generated supervision sequences limit its upper bound. Additionally, the RL-based method IterR fully explores LLM's outputs to achieve better performance, but its overly simplistic reward fails to focus on the final performance, thereby limiting its capabilities.

(5) Integrating dual-modality scoring (Eq. 9) and hybrid rewards designed (Eq. 17), our method R-Mix achieves SOTA accuracy: 90.62% (CR), 90.23% (MR), 91.02% (SST2). The improvements over baselines demonstrate the necessity of joint input-label scoring and balanced reward design.

5 Discussions and Analysis

5.1 Ablation Studies

To further evaluate the key components of R-Mix, we conducted a series of ablation studies.

5.1.1 Impact of Outcome Reward

We analyzed the importance of the outcome reward $\mathcal{R}_{\text{or}}$ by retaining only this reward during the training process. As shown in Table 2, retaining only the outcome reward led to a decline in performance.

Method	CR	MR	SST2
only $\mathcal{R}_{\text{or}}$	0.8704	0.8711	0.8867
only $\mathcal{R}_{\text{infl}}$	0.8867	0.8906	<u>0.9045</u>
only $\mathcal{R}_{\text{entr}}$	0.8672	<u>0.8930</u>	0.7930
only π_{input}	<u>0.8984</u>	0.8642	0.8703
only π_{label}	0.8139	0.7969	0.7487
R-Mix	**0.9062**	**0.9023**	**0.9102**

Table 2: Ablation studies of R-Mix.

This is because the outcome reward is sparse; one can only obtain it after fully selecting K demonstrations. This sparsity of the reward necessitates the introduction of process rewards to ensure that the reward can be distributed across each step of demonstration selection.

5.1.2 Impact of Process Reward

We evaluated the individual impacts of each process reward (influence improvement and label entropy) by examining their contributions to the final accuracy. As shown in Table 2, retaining either reward alone resulted in a performance decline. On the CR and SST2 datasets, retaining only $\mathcal{R}_{\text{infl}}$ led to higher final accuracy than retaining only $\mathcal{R}_{\text{entr}}$, especially on the SST2 dataset. This observation aligns with what we presented in Figures 2(a) and 2(b), where an increase in influence generally leads to an increase in accuracy, while an increase in label entropy does not necessarily guarantee an improvement in accuracy. In other words, optimizing based solely on label entropy as a reward may not necessarily lead to an improvement in the final results, which is consistent with the findings in the table. We need to combine the process reward and the outcome reward to achieve better performance by leveraging their respective advantages.

5.1.3 Impact of Dual-Head Policy

We additionally analyzed our designed dual-head policy, which separately calculates the contribution values of the input and the label in a demonstration and then sums them up. In this part, we separated the scores of the two, i.e., retaining only Score_x or Score_y, denoted as 'only π_{input}' and 'only π_{label}' respectively. As shown in Table 2, retaining the contribution value of either part alone led to a performance decline. Moreover, on the three datasets, retaining only the label contribution resulted in lower performance than retaining only the input score. This result is quite intuitive: in typical tasks, the input contains more information than the label.

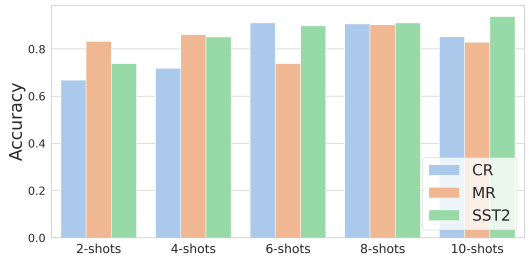


Figure 3: Impact of number of demonstrations.

Training	Testing		
	CR	MR	SST2
CR	**0.9062**	0.8555	0.8281
MR	0.7969	**0.9023**	0.8320
SST2	0.7266	0.7773	**0.9102**

Table 3: Cross-dataset generalization performance.

Especially in our task, the label has only two values (positive and negative), and it is only more relevant to $\mathcal{R}_{\text{entr}}$. However, without the input information, $\mathcal{R}_{\text{or}}$ and $\mathcal{R}_{\text{infl}}$ have no discrimination. This also proves the importance of our dual-head policy in decoupling the contributions of the input and the label respectively.

5.2 Impact of Number of Demonstrations

We study how the number of demonstrations affects R-Mix's performance by experimenting with different shot settings (i.e., 2-shots$\sim$10-shots). As shown in Figure 3, before 8-shots, accuracy generally increases with the number of shots, reflecting the positive impact of more demonstrations on model performance. However, at 10 shots, the performance on CR and MR declines, possibly due to noisy or redundant examples diluting the model's learning signal. To ensure a fair comparison and balance between demonstration sufficiency and quality, we select 8-shots as the default setting.

5.3 Cross-Dataset Generalization Performance

To validate R-Mix's generalizability beyond task-specific specialization, we conduct cross-dataset evaluation where models are trained on one dataset and tested on others. As shown in Table 3, our approach maintains competitive performance under this challenging setting: it surpasses all baselines on the MR dataset (0.8555 accuracy) and ranks second only to DEmO on SST2 (0.8320 vs. DEmO's 0.8711), demonstrating effective knowledge transfer. This may be attributed to our label entropy reward being task-agnostic (i.e., independent of x_{test} in Eq.(13) and Eq.(14)). These results confirm our algorithm's robustness against distribution shifts while preserving task-agnostic adaptability. Furthermore, our approach remains extensible in this regard — any task-agnostic generic list-wise metric can be seamlessly incorporated into our framework, ensuring the generalizability of trained policies.

Method	CR	MR	SST2
Similarity	1.81	1.57	1.57
Best-of-N	46.52	52.19	33.21
MDL	119.72	141.21	96.38
MI	47.25	56.87	36.41
GLE	47.68	55.69	35.86
DEmO	58.82	68.30	54.27
EPR	1.83	1.45	1.85
LLM-R	1.79	1.78	1.57
Se^2	4.13	3.67	4.67
IterR	1.70	1.54	1.73
R-Mix	1.72	1.57	1.54

Table 4: Demonstration sequence construction time(s).

5.4 Inference Efficiency

As shown in Table 4, inference latency is primarily determined by demonstration sequence construction time, as LLM's inference time remains consistent across methods. Ranking by similarity achieves a low latency through heuristic-based point-wise selection. Sample-then-select methods (Best-of-N to DEmO) incur 33.21–141.21s overhead due to exhaustive permutation evaluations using LLM-based list-wise metrics. Crucially, learning-based approaches decouple training and inference: although they require LLM feedback for reward calculation during training, they eliminate LLM dependency during deployment, significantly reducing construction time. IterR and R-Mix achieve latency close to that of similarity-based methods on all tasks, demonstrating their advantages in practical deployment. Se^2 has a longer inference time compared to R-Mix and IterR because it requires step-by-step context concatenation and encoding, whereas our method replaces this with GRU state transitions.

6 Conclusion

We propose a reinforcement learning method R-Mix for ICL, modeling iterative demonstration selection as an MDP. By integrating hybrid rewards combining outcome and process signals, we ad-

dress oversimplified and misguided rewards in existing methods. Analysis shows a positive yet trade-off between process and outcome rewards, highlighting their joint necessity. A dual-head policy architecture enhances performance by decoupling input relevance and label compatibility. Experiments across NLP benchmarks demonstrate superior performance over state-of-the-art methods.

Limitations

Our method R-Mix has several limitations. First, the hybrid reward mechanism relies on linear fusion, leaving room for more sophisticated methods like attention-based weighting. Second, the dual-head policy network directly sums input-semantic and label-content scores, which could be improved by using a gating network for adaptive aggregation. Finally, our on-policy training samples a subset of the dataset, whereas off-policy training with mini-batches could better leverage the full training set for improved generalization. Addressing these limitations could further enhance the framework's performance and applicability.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 62376275, 62472426). Work partially done at Beijing Key Laboratory of Research on Large Models and Intelligent Governance, and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE. Supported by fund for building world-class universities (disciplines) of Renmin University of China.

References

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.

Sirui Chen, Yuan Wang, Zijing Wen, Zhiyu Li, Changshuo Zhang, Xiao Zhang, Quan Lin, Cheng Zhu, and Jun Xu. 2023. Controllable multi-objective re-ranking with policy hypernetworks. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3855–3864.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2022. On the relation between sensitivity and accuracy in in-context learning. *arXiv preprint arXiv:2209.07661*.

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

Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. 2022. Exploration-guided reward shaping for reinforcement learning under sparse rewards. *Advances in Neural Information Processing Systems*, 35:5829–5842.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904.

Joshua Hare. 2019. Dealing with sparse rewards in reinforcement learning. *arXiv preprint arXiv:1910.09281*.

Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. 2023. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19485–19494.

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2024. Efficiently learning at test-time: Active fine-tuning of llms. *arXiv preprint arXiv:2410.08020*.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22.

Jia Li, Yunfei Zhao, Yongmin Li, Ge Li, and Zhi Jin. 2023. Towards enhancing in-context learning for code generation. *arXiv preprint arXiv:2303.17780*.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. se^2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5262–5284.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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

Wenhan Liu, Yutao Zhu, and Zhicheng Dou. 2024b. Demorank: Selecting effective demonstrations for large language models in ranking task. *arXiv preprint arXiv:2406.16332*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Abhishek Naik, Yi Wan, Manan Tomar, and Richard S Sutton. 2024. Reward centering. *arXiv preprint arXiv:2405.09999*.

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

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96.

Chenglei Shen, Xiao Zhang, Teng Shi, Changshuo Zhang, Guofu Xie, and Jun Xu. 2024. A survey of controllable learning: Methods and applications in information retrieval. *arXiv preprint arXiv:2407.06083*.

Suzanna Sia and Kevin Duh. 2023. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. *arXiv preprint arXiv:2305.03573*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Association for Computational Linguistics.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862. Association for Computational Linguistics.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2023. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Zhongxiang Sun, Kepu Zhang, Haoyu Wang, Xiao Zhang, and Jun Xu. 2024. Effective in-context example selection through data compression. *arXiv preprint arXiv:2405.11465*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456, Singapore. Association for Computational Linguistics.

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

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023b. Complementary explanations for effective in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4469–4484.

Changshuo Zhang, Sirui Chen, Xiao Zhang, Sunhao Dai, Weijie Yu, and Jun Xu. 2024. Reinforcing long-term performance in recommender systems with user-oriented exploration policy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1850–1860.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022a. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36:17773–17794.