Bridging Risk Approximation Gaps in Model Predictive Task Sampling via In-Context Modeling

Jiarong Wen[1†], Qi Tao[1†], Kaiyu Zhang[1†], Yun Qu[2], Lizhou Cai[2], Heming Zou[2], Yuhang Jiang[2], Qi Wang[2*]
[1]College of Science, National University of Defense Technology
[2]Department of Automation, Tsinghua University

Abstract

Reinforcement learning (RL) is essential for adaptive decision-making, e.g., robust Meta-RL and domain randomization (DR), and efficient post-training of large language models (LLMs), where tasks are commonly structured as Markov decision processes or prompts. A central bottleneck in all these settings is the cost of agent-environment interactions required for task selection and policy optimization. Model predictive task sampling (MPTS) has emerged as a promising approach to improve sample efficiency by querying informative tasks via a risk predictive model (RPM) trained on optimization history. However, existing RPMs suffer from two failure modes: (i) sparse optimization histories, which limit RPMs' generalization across the task space, and (ii) a one-step lag in risk approximation, which causes biased difficulty estimation as the policy evolves. This work formalizes these failure modes, derives lagged-risk decompositions that expose the resulting temporal bias, and recasts risk prediction as an in-context sequence modeling problem. The developed RPM integrates a one-step look-ahead mechanism with a temporal-weighted sliding window to mitigate data non-stationarity and scarcity, without requiring additional rollouts. Empirically, our approach yields more reliable difficulty estimates and consistent performance gains across robust Meta-RL, DR, and prompt curriculum RL post-training of LLMs.

1 Introduction

Over the past decade, deep reinforcement learning (RL) has achieved remarkable progress in complex sequential decision-making [1, 2, 3, 4, 5]. This work focuses on two settings where RL operates over structured task distributions. (i) In robotics and physical control, agents must adapt to new Markov decision processes (MDPs) in zero-shot or few-shot scenarios under safety constraints [6, 7]. Robust domain randomization (DR) [8] and robust meta-reinforcement learning (Meta-RL) [9] address this by optimizing over MDP distributions and prioritizing risky environments. (ii) In foundation model post-training, RL has become central to incentivizing reasoning in large language models (LLMs) [10, 11], where identifying informative prompts and rollouts is critical for stable optimization [12, 13, 14, 15]. Optimization over prompts further requires policies to explore complex logical search spaces through long-horizon token generation.

Evaluation Bottlenecks in Task Prioritization. Despite their different objectives, both settings share the same bottleneck: expensive policy evaluation. In robust DR and Meta-RL, assessing task risk requires extensive agent-environment trajectories on selected tasks [16, 17, 18, 19]; in LLM post-training, each update requires long Chain-of-Thought (CoT) rollouts [20], making prompt

*Corresponding Author: Qi Wang cheemswang@mail.tsinghua.edu.cn. Equal Contribution†.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

difficulty evaluation costly [12, 13, 21]. Thus, exhaustive task evaluation is impractical, motivating efficient task prioritization.

Conditional Agent-Environment Interactions via Model Predictive Strategies. Model predictive task sampling (MPTS) [22] and its variants [13, 14, 23, 24] reduce unnecessary interactions by using optimization histories, e.g., task or prompt identifiers paired with losses, to train an RPM that approximates task risk online. This enables *conditional interactions*: policy updates target selected tasks without exhaustive rollouts. However, RPM fidelity remains underexplored despite directly governing task selection and policy evolution. Vanilla RPMs [22, 23] suffer from two key limitations: (i) *temporal lag bias*, where outdated policy snapshots misrepresent the current risk landscape; and (ii) *sparse-history generalization*, where limited interaction data restricts generalization across the task space. Analogous issues appear in prompt curriculum learning for LLM post-training, e.g., PCL [14], where temporal lag and sparse evaluations distort curricula and reduce sample efficiency.

Enhancing Risk Predictive Models via In-Context Sequence Modeling. We analyze vanilla RPMs and quantify extrapolation errors caused by temporal lag and data scarcity under small task batches. To address these issues, we propose two remedies: (i) a *one-step look-ahead mechanism* that conditions risk prediction on recent optimization signals to reduce extrapolation bias; and (ii) a *temporal-weighted sliding window* that augments training with recent task trajectories while exponentially downweighting stale data. Together, they instantiate an in-context model of the evolving task risk function, improving prediction under non-stationary policy dynamics with negligible additional rollout cost.

Primary Contributions and Empirical Findings. Focusing on RPM design for the MPTS family, our contributions are two-fold:

1. **Theoretical Analysis of MPTS Vulnerabilities:** We derive a diagnostic lagged-risk decomposition that exposes finite-sample reconstruction error and a policy-update-induced $\mathcal{O}(\eta)$ mismatch, explaining why interpolation-based RPMs can destabilize task-risk rankings.
2. **In-Context Modeling of the Evolving Risk Function:** We introduce in-context RPM (ICRPM) to characterize optimization histories, with one-step look-ahead prediction as the core design and a temporal-weighted sliding window as lightweight data reuse to reduce extrapolation bias and alleviate data scarcity without modifying the policy optimization loop.

Experiments within the MPTS framework show that ICRPM improves sample efficiency and tail-risk robustness on robust DR and Meta-RL benchmarks, and accelerates curriculum-based RL post-training for LLMs, demonstrating better generalization across continuous-control and LLM reasoning tasks.

2 Research Background

Notation and Setup. We consider a unified task space $\mathcal{T}$ that encompasses both continuous control and LLM post-training regimes. Each task $\boldsymbol{\tau} \in \mathcal{T}$ is formalized as a MDP [1]. In DR and Meta-RL, $\boldsymbol{\tau}$ is represented as a continuous descriptor in $\mathbb{R}^d$, whereas in reinforcement learning with verifiable rewards (RLVR), $\boldsymbol{\tau}$ corresponds to a discrete reasoning prompt. At optimization step t, the agent is parameterized by $\boldsymbol{\theta}_t \in \Theta$, representing either a control policy or an autoregressive model $\pi_{\boldsymbol{\theta}_t}$.

Over the course of training adaptive decision-makers or RL finetuning LLMs, we sample a batch of tasks $\mathcal{T}_t^B = \{\boldsymbol{\tau}_{t,i}\}_{i=1}^B \subset \mathcal{T}$ for the agent to interact. Let $\mathcal{L} : \Theta \times \mathcal{T} \to \mathbb{R}$ denote the policy-dependent risk, and define $\ell_t(\boldsymbol{\tau}) := \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\tau})$ as the risk of policy $\pi_{\boldsymbol{\theta}_t}$ on task $\boldsymbol{\tau}$ at iteration t. Concretely, $\ell_t(\boldsymbol{\tau})$ can be the negative episodic return in control settings, $\ell_t(\boldsymbol{\tau}) = -\mathbb{E}_{\pi_{\boldsymbol{\theta}_t}}\left[\sum_{i=0}^{H-1} \bar{\lambda}^i r_{t,i}^{\boldsymbol{\tau}}\right]$ under post-adapted policies $\pi_{\boldsymbol{\theta}_t}$, or the expected penalty in RLVR, e.g., $\ell_t(\boldsymbol{\tau}) = 1 - \mathbb{E}[r_t^{\boldsymbol{\tau}}]$. In the MPTS setup [22, 23, 13, 14], the per-iteration optimization batch is $H_t = \{(\boldsymbol{\tau}_{t,i}, \ell_t(\boldsymbol{\tau}_{t,i}))\}_{i=1}^B$, and the accumulated form $H_{0:t} = \bigcup_{k=0}^t H_k$ tracks the temporal optimization history. We further denote the space of complete historical trajectories by $\mathcal{H}$ and define the RPM as a functional $\hat{\ell}_t : \mathcal{T} \times \mathcal{H} \to \mathbb{R}$, which approximates the non-stationary task risk landscape conditioned on $H_{0:t}$.

General MPTS Pipeline. MPTS [22, 23, 14] executes a *predict-then-select* pipeline, which can instantiate both active prompt selection and robust task sampling. For prompt curriculum learning in RLVR [13, 14], the RPM predicts the sampled success rate $\hat{\rho}^t_{\boldsymbol{\tau}}$, and prompts with intermediate difficulty, i.e., those close to a target threshold ρ^*, typically around 0.5, are preferred [25]. For robust task sampling, the same rule is applied to the predicted risk by replacing $\hat{\rho}^t_{\boldsymbol{\tau}}$ with $\hat{\ell}_t(\boldsymbol{\tau})$ and setting $\rho^* = \max_{\boldsymbol{\tau}\in\mathcal{T}_t^{\hat{B}}} \hat{\ell}_t(\boldsymbol{\tau})$, which encourages near-worst-case task selection.

At each iteration, MPTS proceeds in three steps: ① *Online RPM update.* The RPM is refined with the latest history H_{t-1} by maximizing the likelihood of risk prediction [14] or its approximation [22], e.g., the ELBO in Eq. (2). ② *Amortized evaluation.* Let $\hat{B}$ denote candidate-pool size, a candidate pool $\mathcal{T}_t^{\hat{B}} \sim p(\boldsymbol{\tau})$ is sampled, and the forward passes through the RPM produce predictive risk for all candidate tasks. ③ *Subset selection.* The training task batch is constructed by selecting B tasks with predicted risk closest to the target threshold:

$$\text{Amortized Train RPM:} \quad \max_{\boldsymbol{\psi}\in\Psi,\,\boldsymbol{\phi}\in\Phi} \mathcal{L}_{\text{RPM}}(\boldsymbol{\psi},\boldsymbol{\phi};H_{t-1}), \tag{1a}$$

$$\text{Online Task Evaluation:} \quad p_{\boldsymbol{\psi}}(\ell \mid \hat{\boldsymbol{\tau}}_i, H_{t-1};\boldsymbol{\theta}_t) \xrightarrow{\text{MC}} \left\{\hat{\rho}^t_{\hat{\boldsymbol{\tau}}_i},\, m(\ell_i),\, \sigma(\ell_i)\right\}_{i=1}^{\hat{B}}, \tag{1b}$$

$$\text{Active Subset Selection:} \quad \mathcal{T}_t^B = \arg\text{top-}B_{\boldsymbol{\tau}\in\mathcal{T}_t^{\hat{B}}} \left(-\left\|\hat{\rho}^t_{\boldsymbol{\tau}} - \rho^*\right\|_2^2\right) \tag{1c}$$

In Eq. (1b), $m(\ell_i)$ and $\sigma(\ell_i)$ are the predictive mean and standard deviation of the task outcome estimated via Monte Carlo sampling, which are then mapped to the acquisition scalar $\hat{\rho}^t_{\hat{\boldsymbol{\tau}}_i}$ to quantify task priority. With the queried batch in Eq. (1c), the policy model $\pi_{\boldsymbol{\theta}}$ is accordingly updated in the next iteration, e.g., $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\frac{1}{B}\sum_{b=1}^{B}\nabla_{\boldsymbol{\theta}}\ell_t(\boldsymbol{\tau}_b)$ in DR. By repeating these steps for T rounds, MPTS achieves either prompt curriculum learning in LLMs or robust Meta-RL and DR.

RPM and Variational Inference. Existing MPTS frameworks typically instantiate the RPM as a VAE-based latent generative predictor [26], where a latent state $\mathbf{z}_t$ summarizes historical task-risk observations. Using streaming variational inference [27, 28], the RPM maximizes the per-step ELBO:

$$\mathcal{G}_{\text{ELBO}}(\boldsymbol{\psi},\boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_t|H_t)}\left[\sum_{i=1}^{B}\log p_{\boldsymbol{\psi}}(\ell_t(\boldsymbol{\tau}_{t,i})|\boldsymbol{\tau}_{t,i},\mathbf{z}_t)\right] - \beta\text{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}_t|H_t)||q_{\bar{\boldsymbol{\phi}}}(\mathbf{z}_t|H_{t-1})\right), \tag{2}$$

where $\bar{\boldsymbol{\phi}}$ represents the parameters from the previous iteration, kept fixed to provide a stable prior. For task evaluation, RPMs use Monte Carlo forward passes to estimate $p(\ell|\boldsymbol{\tau},H_{0:t};\boldsymbol{\theta}_t) \approx \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_t|H_t)}\left[p_{\boldsymbol{\psi}}(\ell|\boldsymbol{\tau},\mathbf{z}_t;\boldsymbol{\theta}_t)\right]$. These predictions are then used by the acquisition rule in Eq. (1c) for robust [22, 23] or curriculum optimization [13, 14]. Thus, vanilla RPMs rely on generalization at $\boldsymbol{\theta}_t$ to approximate task difficulty after the policy update to $\boldsymbol{\theta}_{t+1}$, creating the temporal mismatch analyzed next.

3 From Theoretical Investigation to Practical Enhancement

This section analyzes the theoretical bottlenecks of RPMs, introduces the In-Context Predictive Model (ICRPM), and discusses its implementation within MPTS.

3.1 Vulnerabilities of Vanilla RPM

Although MPTS provides a sample-efficient surrogate for task risk estimation, the vanilla RPM overlooks a common failure mode, i.e., the continual evolution of both the policy and the task risk landscape induces rank-flipping (Fig. 1) in task risk that cannot be corrected post-hoc.

Instability of Task Risk Rankings. MPTS relies on accurate task-risk ranking, but task risk is policy-dependent: after updating from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$, the ordering induced by $\ell_t(\boldsymbol{\tau})$ may

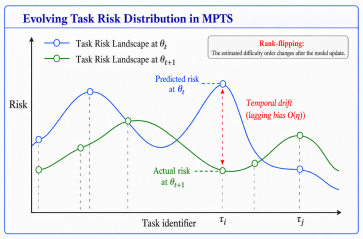


Figure 1: **Task Risk Distribution Shift over Iterations.** Using the risk generalization results at $\boldsymbol{\theta}_t$ tends to cause the *rank-flipping* at $\boldsymbol{\theta}_{t+1}$, i.e., $\text{sgn}[\ell_t(\boldsymbol{\tau}_i) - \ell_t(\boldsymbol{\tau}_j)] \neq \text{sgn}[\ell_{t+1}(\boldsymbol{\tau}_i) - \ell_{t+1}(\boldsymbol{\tau}_j)]$.

differ from that of $\ell_{t+1}(\boldsymbol{\tau})$. As shown in Fig. 1, this policy-induced drift creates a non-stationary difficulty landscape. Prior analysis [22] shows that ranking stability requires sufficiently small learning rates and smooth loss landscapes, conditions often violated in practice. Hence, even a perfectly accurate RPM at $\boldsymbol{\theta}_t$ may fail to preserve the correct ranking at $\boldsymbol{\theta}_{t+1}$, which we call the *task risk distribution shift*.

Lagged Risk Prediction and Reconstruction Bound. The vanilla RPM, a VAE-based risk generative model [22, 23], is trained on the latest history batch H_t of size B in each per-step update, where supervision targets are evaluated at the latest yet lagged parameter $\boldsymbol{\theta}_t$. The induced structured risk distribution shift cannot be resolved by increasing sample size alone: even under standard capacity and smoothness assumptions, the prediction error includes a temporal mismatch term induced by the policy update from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$.

Theorem 3.1 (Per-step Lagged Prediction Bound for Vanilla RPMs)**.** *Assume that $\ell(\boldsymbol{\tau}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\tau}) \in [0, M]$ for all $\boldsymbol{\theta} \in \Theta$, $\boldsymbol{\tau} \in \mathcal{T}$, and that $\ell(\boldsymbol{\tau})$ is L-Lipschitz in $\boldsymbol{\theta}$. The policy update uses a normalized average loss computed by i.i.d. samples and $\|\nabla_{\boldsymbol{\theta}} \ell_t(\boldsymbol{\tau}_i)\| \le G$. Let $\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau})$ denote the predictive risk function induced by the vanilla RPM at iteration t. Define $\epsilon_{\text{app}}^{\text{rec}} := \inf_{\boldsymbol{\psi},\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{\tau}\sim p}\big[\big(\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau}) - \ell_t(\boldsymbol{\tau})\big)^2\big]$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of H_t,*

$$\mathbb{E}_{\boldsymbol{\tau}\sim p}\big[|\hat{\ell}_t(\boldsymbol{\tau}) - \ell_{t+1}(\boldsymbol{\tau})|\big] \le \underbrace{\sqrt{\epsilon_{\text{app}}^{\text{rec}} + \epsilon_{\text{gen}}^{\text{rec}}(B,\delta)}}_{\text{①Reconstruction Approximation Error}} + \underbrace{L\eta G}_{\text{②Temporal Lagging Error}}. \tag{3}$$

If the reconstruction loss class $\mathcal{G}^{\text{rec}} = \{(\boldsymbol{\tau}, y) \mapsto (\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau}) - y)^2 : \boldsymbol{\psi} \in \Psi, \boldsymbol{\phi} \in \Phi\}$ has Rademacher complexity $\mathfrak{R}_B(\mathcal{G}^{\text{rec}})$ and is bounded by M^2, then $\epsilon_{\text{gen}}^{\text{rec}}(B,\delta) = 4\mathfrak{R}_B(\mathcal{G}^{\text{rec}}) + 2M^2\sqrt{\frac{\ln(2/\delta)}{2B}}$.

Theorem 3.1 decomposes the lagged prediction error of vanilla RPMs into ① finite-sample reconstruction and approximation error from sparse optimization histories, and ② policy-update-induced temporal mismatch from applying risk estimates at $\boldsymbol{\theta}_t$ to task selection under $\boldsymbol{\theta}_{t+1}$. Thus, interpolation-based vanilla RPMs [22, 14] incur an $\mathcal{O}(\eta)$ lagging bias, which may distort task-risk rankings and weaken task-space generalization.

3.2 Context-based Extrapolative Risk Prediction

Having identified temporal misalignment and data scarcity as the main limitations, we introduce ICRPM with two components: (i) *a look-ahead mechanism* that conditions risk prediction on recent optimization context; and (ii) *temporal-weighted trajectory augmentation* with exponential decay weighting to improve efficiency under tight sampling budgets.

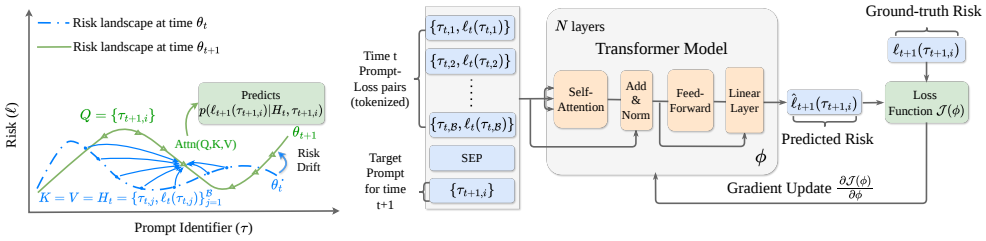


Figure 2: **Left**: The risk landscape evolves from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$, with historical prompt-loss pairs H_t modeling temporal risk dynamics and predicting future risk using an attention mechanism. **Right**: The Transformer-based ICRPM tokenizes prompt-loss sequences with a target prompt at time $t+1$, and a regression head estimates the future risk $\hat{\ell}_{t+1}$ by minimizing the discrepancy with ground-truth ℓ_{t+1} through gradient optimization.

In-Context Look-Ahead Risk Estimation. To overcome the single-step temporal misalignment in interpolation-based evaluation, ICRPM adopts an extrapolative formulation $p_\phi(\ell_{t+1}(\boldsymbol{\tau}) \mid \boldsymbol{\tau}, H_t)$, where the latest batch H_t serves as the inference-time optimization context. By directly targeting the future policy state $\boldsymbol{\theta}_{t+1}$, ICRPM predicts next-step task risk rather than reconstructing the lagged risk under $\boldsymbol{\theta}_t$, thereby alleviating the $\mathcal{O}(\eta)$ temporal bias identified in Theorem 3.1.

Exponentially Decayed Sliding Window Augmentation. The sliding window is a data-reuse mechanism for training the look-ahead predictor under limited task evaluations. ICRPM maintains the W most recent history batches and trains on consecutive context–target pairs $\left\{\{(\boldsymbol{\tau}_{t-k+1,i}, \ell_{t-k+1}(\boldsymbol{\tau}_{t-k+1,i}), H_{t-k})\}_{i=1}^{B}\right\}_{k=1}^{W}$, weighted by γ^k. This reuse mitigates the sparse-history issue in Theorem 3.1: recent batches provide more training pairs, while exponential decay gives less weight to outdated histories.

Optimization Objective. Combining the look-ahead formulation and historical augmentation, ICRPM minimizes a temporal-weighted negative log-likelihood over in-context prediction targets:

$$\min_{\boldsymbol{\phi}} \mathcal{J}(\boldsymbol{\phi}) = -\sum_{k=1}^{W} \gamma^k \left(\frac{1}{B} \sum_{i=1}^{B} \ln p_{\boldsymbol{\phi}} \left(\ell_{t-k+1}(\boldsymbol{\tau}_{t-k+1,i}) | \boldsymbol{\tau}_{t-k+1,i}, H_{t-k} \right) \right) \tag{4}$$

where $\boldsymbol{\phi}$ denotes model parameters, W is the look-back window and $\gamma \in (0, 1]$ controls temporal decay. The objective in Eq. (4) trains ICRPM on already observed one-step transitions: at round t, it uses consecutive history pairs $H_{s-1} \to H_s$ with $s \leq t$ from the buffer. The unobserved batch H_{t+1} is not used when selecting $\mathcal{T}_{t+1}^{B}$. Only after the selected batch is evaluated, the resulting H_{t+1} is appended to the buffer and used as a target in later RPM updates. At the inference stage, i.e., task evaluation under the coming $\boldsymbol{\theta}_{t+1}$, ICRPM takes in the latest batch H_t as the context set $\mathcal{C}_t$ and the candidate identifier $\boldsymbol{\tau} \in \mathcal{T}_{t+1}^{\hat{B}}$ to predict risk $\hat{\ell}_{t+1}(\boldsymbol{\tau})$.

Algorithm 1: ICRPM Training & Prompt and Robust Predictive Sampling (PPS and RPS)

Input: Task/Prompt prior $p(\boldsymbol{\tau})$; history buffer $\mathcal{S} = \{H_i\}$; $\boldsymbol{\phi}$; B, $\hat{B}$; W; γ; α; `mode` $\in \{\texttt{PPS}, \texttt{RPS}\}$; target ρ^*.
Output: Selected batch $\mathcal{T}_{t+1}^{B}$

`// Online ICRPM Optimization`
for *each new batch* H_t **do**
  `// FIFO`
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{H_t\}$;
  if $|\mathcal{S}| > W + 1$ **then** Drop oldest batch in $\mathcal{S}$;
  $\mathcal{J}(\boldsymbol{\phi}) \leftarrow 0$;
  for $k = 1$ **to** $\min(W, |\mathcal{S}| - 1)$ **do**
    `// Context vs. look-ahead`
    $H_{ctx}, H_{tgt} = H_{t-k}, H_{t-k+1}$;
    $\mathcal{L}_k \leftarrow -\frac{1}{B} \sum_j \ln p_{\boldsymbol{\phi}}(y_{tgt,j} \mid \boldsymbol{\tau}_{tgt,j}, H_{ctx})$;
    $\mathcal{J}(\boldsymbol{\phi}) \mathrel{+}= \gamma^k \mathcal{L}_k$;
  `// Update` $\boldsymbol{\phi}$ `by minimizing` $\mathcal{J}(\boldsymbol{\phi})$
  $\boldsymbol{\phi} = \boldsymbol{\phi} - \alpha \nabla_{\boldsymbol{\phi}} \mathcal{J}(\boldsymbol{\phi})$;

`// Predictive Task/Prompt Selection`
Sample candidate pool:
$\mathcal{T}_{t+1}^{\hat{B}} = \{\hat{\boldsymbol{\tau}}_{t+1,i}\}_{i=1}^{\hat{B}} \sim p(\boldsymbol{\tau})$;
for $i = 1$ **to** $\hat{B}$ **do**
  `// One-step look-ahead`
  $\hat{y}_i^{t+1} = E_{p_{\boldsymbol{\phi}}(y|\hat{\boldsymbol{\tau}}_{t+1,i}, H_t)}[y]$;
if *mode* $=$ *PPS* **then**
  $c_{t+1}^* \leftarrow \rho^*$; `// ŷ is success rate`
else
  $c_{t+1}^* \leftarrow \max_{\boldsymbol{\tau}} \hat{y}_{\boldsymbol{\tau}}^{t+1}$; `// ŷ is risk`
$\mathcal{T}_{t+1}^{B} = \arg\text{top-}B_{\boldsymbol{\tau} \in \mathcal{T}_{t+1}^{\hat{B}}} \left(-\|\hat{y}_{\boldsymbol{\tau}}^{t+1} - c_{t+1}^*\|_2^2 \right)$;
return $\mathcal{T}_{t+1}^{B}$

First-order interpretation. The look-ahead objective in Eq. (4) aligns RPM supervision with the policy state used for task selection. The following proposition gives a lightweight formalization: if recent optimization history enables the in-context predictor to capture the local risk drift, then the next-step prediction error is controlled by a context approximation term and a second-order smoothness residual.

Proposition 3.2 (Look-ahead reduces temporal mismatch)**.** *Let* $\boldsymbol{\Delta}_t = \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$ *with* $\|\boldsymbol{\Delta}_t\| \leq \eta G$, *and assume that* $\ell(\boldsymbol{\theta}, \boldsymbol{\tau})$ *is* β*-smooth in* $\boldsymbol{\theta}$*. For analysis, define the first-order look-ahead risk as* $m_t(\boldsymbol{\tau}) = \ell_t(\boldsymbol{\tau}) + \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_t, \boldsymbol{\tau})^\top \boldsymbol{\Delta}_t$*. If the ICRPM predictor satisfies* $\mathbb{E}_{\boldsymbol{\tau} \sim p} |\hat{\ell}_{t+1}^{\mathrm{IC}}(\boldsymbol{\tau}) - m_t(\boldsymbol{\tau})| \leq \epsilon_{\mathrm{IC}}$*, then*

$$\mathbb{E}_{\boldsymbol{\tau} \sim p} \left[|\hat{\ell}_{t+1}^{\mathrm{IC}}(\boldsymbol{\tau}) - \ell_{t+1}(\boldsymbol{\tau})| \right] \leq \epsilon_{\mathrm{IC}} + \frac{\beta}{2} \eta^2 G^2. \tag{5}$$

This result provides a first-order explanation for the advantage of look-ahead risk prediction. A lagged RPM estimates ℓ_t for task selection under $\boldsymbol{\theta}_{t+1}$ and therefore retains the $O(\eta)$ temporal mismatch identified in Theorem 3.1. In contrast, ICRPM directly targets the next-step risk: when the context predictor captures the local risk drift, the leading lagging bias is replaced by the in-context approximation error ϵ_{IC} and a second-order residual.

3.3 Implementation and Neural Architecture

Algorithm 1 summarizes the implementation; the selected batch is passed to downstream optimizers such as robust Meta-RL, DR, or LLM RLVR. ICRPM performs the extrapolative formulation $p(\ell|\tau, H_t; \boldsymbol{\theta}_{t+1})$. At each iteration, the context set $\mathcal{C}_{t-1} = \{(\boldsymbol{\tau}_i, \ell_{t-1}(\boldsymbol{\tau}_i))\}_{i=1}^{B}$ consists of task-risk pairs evaluated under $\boldsymbol{\theta}_{t-1}$, while each training target is a pair $(\boldsymbol{\tau}, \ell_t(\boldsymbol{\tau}))$ under the last policy state $\boldsymbol{\theta}_t$. This yields B supervised training episodes per iteration, one for each $j \in [B]$, without any additional environment interaction.

Risk Sequence Tokenization and Cross-Attention. ICRPM has a simple design. As illustrated in Fig. 2, we tokenize the task-risk pairs and adopt the Transformer as the default neural architecture to capture the correlation between the target and past risk sequences. This way encodes the context into a permutation-invariant representation and produces $p(\ell|\tau, H_t; \boldsymbol{\theta}_{t+1})$ via cross-attention.

4 Results and Analysis

This section empirically examines ICRPM's influence within MPTS. Motivated by the failure modes diagnosed in Section 3.1, the experiments are designed to answer two Research Questions (RQs):

1. Do extrapolative RPMs enhance task evaluation reliability over iteration?
2. How does task evaluation reliability influence performance in downstream applications?

We track Spearman Rank Correlation (SRC) between predicted task difficulty and ground-truth task risk as a diagnostic measure of rank reliability, and report CVaR_α [29] for robust optimization scenarios and test average performance for RLVR. Full experimental details are provided in Appendix E.

Benchmarks and Baselines. We evaluate ICRPM on few-shot sinusoid regression, Meta-RL, and DR, and LLM post-training via RLVR. (i) For regression and continuous-control, we retain the standard MPTS protocol [22], e.g., the same task distribution, batch size, candidate pool, selection budget, and downstream optimizer; and baselines include ERM [30], GDRM [31], DRM [17, 32], OHTM [33], MPTS [22]. (ii) For LLM RL post-training, we adopt the curriculum-based setup of [14], and baselines are uniform sampling and SoTA predictive sampling methods such as PCL [14], GRESO [34], and MoPPS [13]. For both cases, ICRPM replaces only the interpolation-based RPM with a parameter-matched context-conditioned one-step extrapolative predictor.

4.1 Robust Few-Shot Sinusoid Regression

We first validate ICRPM on the classic few-shot sinusoid regression benchmark [9], which provides a controlled setting to isolate the effect of improved task risk estimation. Tasks are parameterised by amplitude and phase, and MAML is used as the meta-learning backbone.

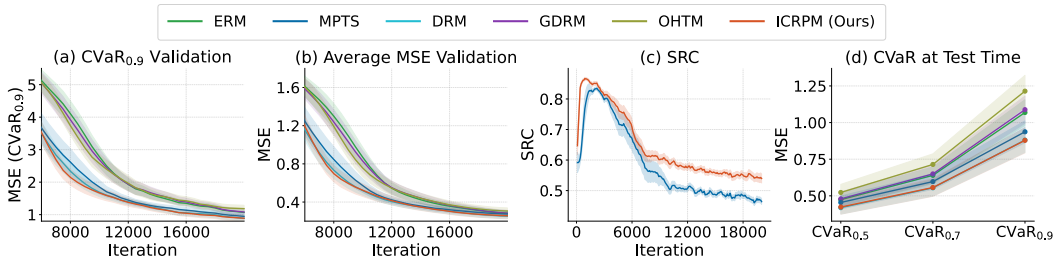


Figure 3: **Few-shot sinusoid regression results (10 Runs, Mean $\pm$ Standard Error of the Mean (SEM)).** From left to right: (a) validation $\text{CVaR}_{0.9}$ MSE curves during meta-training; (b) validation average MSE curves during meta-training; (c) SRC of predicted vs. ground-truth difficulty during training; (d) meta-testing CVaR_α MSE at multiple robustness levels. ICRPM (ours) achieves consistently lower tail-risk error and maintains higher SRC throughout training.

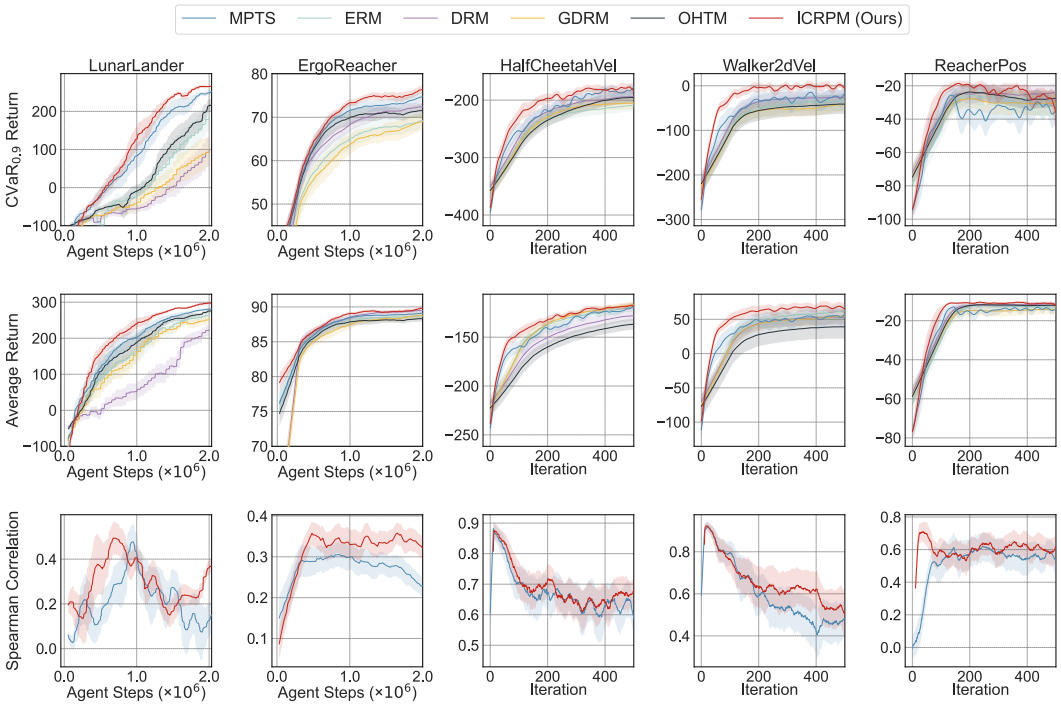


Figure 4: **DR (columns 1–2) and Meta-RL (columns 3–5) learning curves and correlation trends (10 Runs, Mean ± SEM).** Rows from top to bottom: $\text{CVaR}_{0.9}$ returns, average returns, and SRC of predicted vs. ground-truth task risk. ICRPM consistently attains the highest tail-risk performance while maintaining competitive average returns, alongside superior rank correlation throughout training.

Few-Shot Regression Analysis. In Fig. 3, ICRPM consistently outperforms all baselines. It achieves lower average MSE (Fig. 3b) and $\text{CVaR}_{0.9}$ MSE (Fig. 3a) with faster convergence, while maintaining higher SRC with ground-truth task risk throughout training (Fig. 3c). At meta-test time (Fig. 3d), ICRPM achieves the lowest CVaR_α MSE across all $\alpha \in \{0.5, 0.7, 0.9\}$, suggesting that more reliable task prioritisation is accompanied by stronger tail-task generalization. Vanilla MPTS underperforms in both convergence stability and worst-case performance.

4.2 Robust Meta-RL and Domain Randomization

We evaluate ICRPM on Meta-RL over three MuJoCo environments [9, 35] and zero-shot DR on Ergo-Reacher and Lunar-Lander [36]. Both settings require generalisation across a distribution of MDPs parameterised by continuous task identifiers, where robustness to tail or out-of-distribution tasks is critical.

Table 1: **Evaluation return on five robotic control tasks. (10 Runs, Mean ± SEM)** The best results are bold, with the runner-up underlined.

Method	**HalfCheetahVel**		**Walker2dVel**		**ReacherPos**		**LunarLander**		**ErgoReacher**	
	$\text{CVaR}_{0.5}$	$\text{CVaR}_{0.9}$	$\text{CVaR}_{0.5}$	$\text{CVaR}_{0.9}$	$\text{CVaR}_{0.5}$	$\text{CVaR}_{0.9}$	$\text{CVaR}_{0.5}$	$\text{CVaR}_{0.9}$	$\text{CVaR}_{0.5}$	$\text{CVaR}_{0.9}$
ERM [30]	-153.6±3.9	-209.9±7.6	23.5±15.6	-34.2±20.7	-17.5±1.9	-31.1±5.4	243.0±14.2	185.5±30.4	83.8±0.7	70.1±1.4
DRM [17, 32]	-151.1±6.2	-195.5±8.4	<u>26.1</u>±12.2	<u>-26.0</u>±14.7	<u>-16.0</u>±2.2	**-26.5**±5.6	184.2±25.0	96.8±36.6	<u>85.3</u>±0.4	72.9±1.0
GDRM [31]	-149.8±4.3	-206.4±8.0	16.1±13.3	-42.8±16.6	-19.6±4.3	-30.7±6.1	207.9±18.4	93.7±38.0	84.4±0.7	70.2±1.7
OHTM [33]	-160.0±7.3	-194.8±10.3	7.6±21.5	-40.8±21.3	-16.3±1.4	-28.9±5.6	258.7±6.4	215.7±17.9	83.8±0.7	72.6±2.0
MPTS [22]	<u>-143.6</u>±4.6	<u>-184.0</u>±8.3	24.6±16.2	-27.9±19.1	-20.1±3.4	-33.2±8.0	<u>268.5</u>±3.4	<u>251.8</u>±4.1	84.7±0.5	<u>75.7</u>±1.1
ICRPM (Ours)	**-140.5**±3.5	**-180.2**±6.4	**43.2**±6.7	**-1.5**±7.2	**-15.2**±0.9	<u>-27.8</u>±3.8	**285.3**±2.6	**265.8**±2.1	**86.5**±0.4	**78.3**±0.7

Meta-RL Analysis. Fig. 4 (columns 3–5) reports training dynamics on HalfCheetahVel, Walker2dVel, and ReacherPos. ICRPM achieves the highest $\text{CVaR}_{0.9}$ validation returns on all three benchmarks while preserving competitive average performance — a balance DRM fails to maintain, notably degrading on Walker2dVel. The SRC curves (row 3) confirm that the Transformer-based RPM with temporal-weighted augmentation sustains markedly higher rank correlation than MPTS throughout training. At meta-test time (Table 1), ICRPM attains the best CVaR_α returns across nearly all environments and robustness levels, with especially large gains on Walker2dVel ($\text{CVaR}_{0.9}$: -1.5 vs. -27.9 for MPTS) and substantially reduced variance.

Domain Randomization Analysis. On Ergo-Reacher and Lunar-Lander (Fig. 4, columns 1–2), ICRPM consistently outperforms all baselines in $\text{CVaR}_{0.9}$ returns while remaining competitive in average performance. Notably, on Lunar-Lander ICRPM improves $\text{CVaR}_{0.9}$ at convergence by over 40% relative to ERM, effectively handling the long-tail of adversarial engine-strength configurations. The SRC advantage of ICRPM over MPTS persists in both DR environments, demonstrating that mitigating temporal lagging bias extends beyond episodic few-shot learning to the zero-shot continuous-control regime.

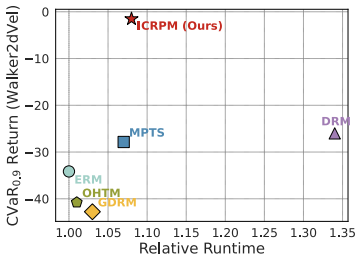


Figure 5: **$\text{CVaR}_{0.9}$ return vs. relative runtime on Walker2dVel.**

Computational Overhead. Fig. 5 plots $\text{CVaR}_{0.9}$ return against relative runtime on Walker2dVel (normalised to ERM). ICRPM incurs only 8% overhead (1.08×), comparable to MPTS (1.07×) and far below DRM (1.34×), whose per-iteration importance-weight computation dominates cost. GDRM and OHTM add minimal time (1.03× and 1.01×) but lag considerably in tail-risk performance. The scatter clearly shows that ICRPM occupies the ideal upper-left region with the highest $\text{CVaR}_{0.9}$ return at near-baseline computational cost, confirming that its performance gains come at negligible additional runtime budget.

4.3 Sample-Efficient RLVR on Countdown

To evaluate the effectiveness of ICRPM in accelerating LLM post-training, we conduct experiments on the Countdown symbolic reasoning task. Countdown requires composing given numbers with arithmetic operations to reach a target value, offering verifiable binary rewards that enable straightforward RLVR training.

Experimental Setup. We use **Qwen3-4B-Base** [37] as the policy backbone. Following the RLVR paradigm, we employ Group Relative Policy Optimization (GRPO) [38] to optimize the model with verifiable rewards that consider both mathematical correctness and format adherence. Training is performed on a 20k-instance subset of Countdown-34 (CD34); evaluation is conducted on both CD34 and the harder Countdown-4 (CD4) variant.

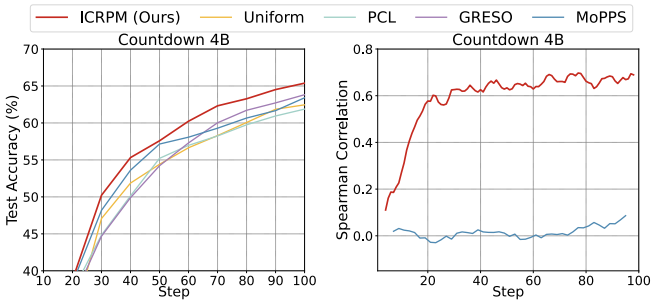


Figure 6: **Test accuracy (left) and SRC (right) on Countdown 4B.** ICRPM demonstrates significantly faster convergence and higher correlation with ground-truth difficulty throughout RL post-training.

Table 2: **Evaluation results on Countdown.** '+' denotes finetuning with the corresponding method. **Bold**: best; underline: runner-up.

Method	CD4	CD34	Avg.
Qwen3-4B [37]	1.3	3.5	2.4
+Uniform	51.1	73.8	62.5
+PCL [14]	51.0	72.8	61.9
+GRESO [34]	53.8	73.8	63.8
+MoPPS [13]	52.9	73.9	63.4
+ICRPM	**55.7**	**75.1**	**65.4**

Curriculum RLVR Analysis. As summarised in Table 2 and Fig. 6, ICRPM consistently outperforms all baselines. On CD4, ICRPM achieves **55.7%** accuracy, surpassing GRESO by 1.9 points and MoPPS by 2.8 points, with similar gains on CD34. The training curve (Fig. 6, left) shows faster convergence and superior sample efficiency, while the SRC curve (right) indicates more reliable online prompt-difficulty rankings as the policy evolves. This addresses the sparse-history and temporal-lag issues discussed in Section 3.1: unlike prompt-specific models such as MoPPS, ICRPM shares information across prompts and better tracks non-stationary difficulty during RL post-training.

4.4 Ablation Studies

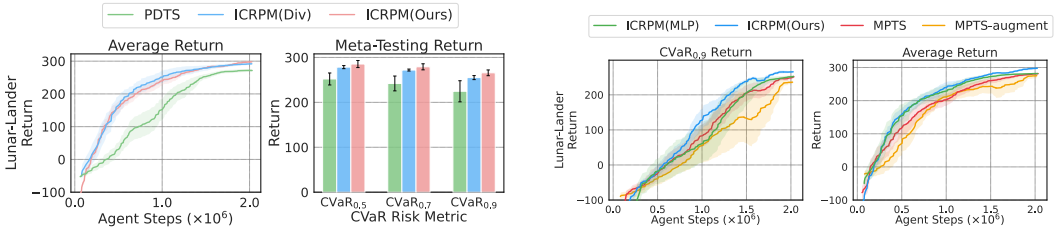


Figure 7: **Ablation studies on Lunar-Lander (10 Runs).** *Left*: Impact of acquisition principles. Two ICRPM variants (worst-case and diversity-aware) versus PDTS [23]. *Right*: Architecture ablation comparing ICRPM with MLP or Transformer backbone, vanilla MPTS, and MPTS with history augmentation.

We ablate the subset acquisition principles, RPM backbones, window size W, and decay factor γ on Lunar-Lander. Fig. 7 shows that ICRPM improves over PDTS under both worst-case and diversity-aware acquisition. For architecture and history augmentation, ICRPM with a parameter-matched MLP still outperforms vanilla MPTS, while MPTS augmented with the same three-step window and $\gamma = 0.9$ remains below ICRPM and ICRPM (MLP). This indicates that the gains mainly come from context-conditioned look-ahead risk prediction, rather than a specific acquisition rule, Transformer backbone, or history-reuse heuristic. Together, these ablations match the decomposition in Theorem 3.1: look-ahead prediction addresses temporal lag, while windowed reuse mitigates sparse history. Fig. 8 shows that W has a stronger effect than γ: we choose $W = 3$ because it gives the best $\text{CVaR}_{0.9}$ performance in the ablation, and use $\gamma = 0.9$ across all benchmarks as a robust default that mildly discounts older histories.

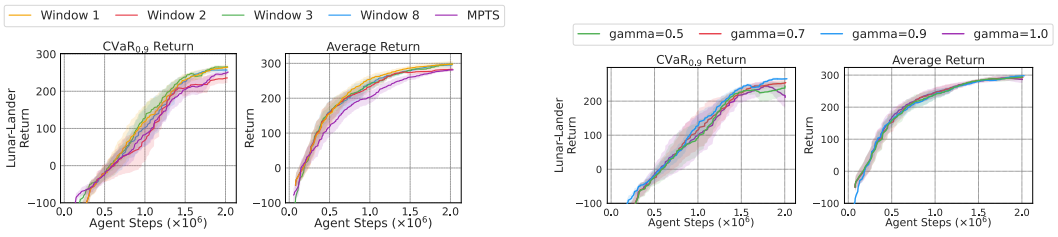


Figure 8: **Sensitivity to W and γ on Lunar-Lander (10 Runs).** *Left*: Sensitivity to look-back window W. $\text{CVaR}_{0.9}$ returns, average returns under $W \in \{1, 2, 3, 8\}$. *Right*: Sensitivity to decay factor γ. $\text{CVaR}_{0.9}$ returns, average returns under $\gamma \in \{0.5, 0.7, 0.9, 1.0\}$.

5 Conclusion

This work identifies a structural limitation of MPTS: vanilla RPMs estimate risks under the current policy but use them for post-update task selection, inducing first-order lagging bias and unstable task rankings. We propose ICRPM, which reframes risk prediction as context-conditioned one-step extrapolation with temporal-weighted history augmentation. Empirically, ICRPM improves risk ranking, sample efficiency, and tail-risk performance across regression, robust Meta-RL, domain randomization, and RLVR-based LLM post-training, suggesting that task samplers should model policy-induced risk dynamics rather than rely on static or lagged difficulty estimates.

Acknowledgment

Dr. Qi Wang acknowledges support from the National Natural Science Foundation of China (NSFC) under the grant number 62306326. The authors declare no competing interests.

References

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[3] Yun Qu, Boyuan Wang, Jianzhun Shao, et al. Hokoff: Real game dataset from honor of kings and its offline reinforcement learning benchmarks. In *NeurIPS*, 2023.

[4] Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *NeurIPS*, 36, 2024.

[5] Qi Wang and Herke Van Hoof. Model-based meta reinforcement learning using graph structured surrogate models and amortized policy search. In *International Conference on Machine Learning*, pages 23055–23077. PMLR, 2022.

[6] Stefano Carpin, Yin-Lam Chow, and Marco Pavone. Risk aversion in finite markov decision processes using total cost criteria and average value at risk. In *2016 ieee international conference on robotics and automation (icra)*, pages 335–342. IEEE, 2016.

[7] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17305–17315, 2022.

[8] Fabio Muratore, Felix Treede, Michael Gienger, and Jan Peters. Domain randomization for simulation-based policy optimization with transferability assessment. In *Conference on Robot Learning*, pages 700–713. PMLR, 2018.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[10] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[11] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[12] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[13] Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1240–1250, 2026.

[14] Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient LLM post-training. In *The Fourteenth International Conference on Learning Representations*, 2026.

[15] Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Weijie Liu, Clive Bai, Kai Yang, Yangkun Chen, Saiyong Yang, et al. Small generalizable prompt predictive models can steer efficient rl post-training of large reasoning models. *arXiv preprint arXiv:2602.01970*, 2026.

[16] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[17] Qi Wang, Yiqin Lv, Zheng Xie, Jincai Huang, et al. A simple yet effective strategy to robustify the meta learning paradigm. *Advances in Neural Information Processing Systems*, 36, 2024.

[18] Yiqin Lv, Cheems Wang, Dong Liang, and Zheng Xie. Theoretical investigations and practical enhancements on tail task risk minimization in meta learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[19] Qi Wang, Yiqin Lv, Yixiu Mao, Yun Qu, Yi Xu, and Xiangyang Ji. Robust fast adaptation from adversarially explicit task distribution generation. In *Proceedings Of The 31st ACM SIGKDD Conference On Knowledge Discovery And Data Mining V. 1*, pages 1481–1491, 2025.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[21] Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Yingyue Li, Wutong Xu, Lizhou Cai, Weijie Liu, Clive Bai, et al. Listwise policy optimization: Group-based rlvr as target-projection on the llm response simplex. *arXiv preprint arXiv:2605.06139*, 2026.

[22] Qi Cheems Wang, Zehao Xiao, Yixiu Mao, Yun Qu, Jiayi Shen, Yiqin Lv, and Xiangyang Ji. Model predictive task sampling for efficient and robust adaptation, 2025.

[23] Yun Qu, Qi Cheems Wang, Yixiu Mao, Yiqin Lv, and Xiangyang Ji. Fast and robust: Task sampling with posterior and diversity synergies for adaptive decision-makers in randomized environments. *arXiv preprint arXiv:2504.19139*, 2025.

[24] Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Dynamics-predictive sampling for active rl finetuning of large reasoning models. In *International Conference on Learning Representations*, volume 2026, pages 25242–25270, 2026.

[25] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning, 2026.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[27] Ankush Ganguly and Samuel W. F. Earp. An introduction to variational inference, 2021.

[28] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[29] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[30] Vladimir Naumovich Vapnik. Statistical learning theory. 1998.

[31] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[32] Ido Greenberg, Shie Mannor, Gal Chechik, and Eli Meirom. Train hard, fight easy: Robust meta reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Ramnath Kumar, Tristan Deleu, and Yoshua Bengio. The effect of diversity in meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8396–8404, 2023.

[34] Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025.

[35] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[36] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *Conference on Robot Learning*, pages 1162–1176. PMLR, 2020.

[37] An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025.

[38] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[39] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: a cvar optimization approach. *Advances in neural information processing systems*, 28, 2015.

[40] Yinlam Chow. *Risk-sensitive and data-driven sequential decision making*. PhD thesis, Stanford University, 2017.

[41] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018.

[42] Qi Tao, Jiarong Wen, Jing Yang, Guanlin Wu, Zhang Kaiyu, Yiqin Lv, Wumei Du, Xingxing Liang, and Qi Wang. Latent space robust optimization of neural processes with aligned stratified order-statistic loss reduction. In *Forty-third International Conference on Machine Learning*, 2026.

[43] He Li, Haoang Chi, Mingyu Liu, Wanrong Huang, Liyang Xu, and Wenjing Yang. Transformer-based spatial-temporal counterfactual outcomes estimation. In *International Conference on Machine Learning*, volume 267, pages 34185–34208, 2025.

[44] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Advances in neural information processing systems*, 28, 2015.

[45] Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8522–8528. IEEE, 2019.

[46] Marc Rigter, Bruno Lacerda, and Nick Hawes. Risk-averse bayes-adaptive reinforcement learning. *Advances in Neural Information Processing Systems*, 34:1142–1154, 2021.

[47] Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient risk-averse reinforcement learning. *Advances in Neural Information Processing Systems*, 35:32639–32652, 2022.

[48] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[49] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

[50] Qi Wang and Herke Van Hoof. Learning expressive meta-representations with mixture of expert neural processes. *Advances in neural information processing systems*, 35:26242–26255, 2022.

[51] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 33:18860–18871, 2020.

[52] Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021.

[53] Gabriele Tiboni, Pascal Klink, Jan Peters, Tatiana Tommasi, Carlo D'Eramo, and Georgia Chalvatzaki. Domain randomization via entropy maximization. *arXiv preprint arXiv:2311.01885*, 2023.

[54] Wenjing Yang, Haoang Chi, Yibing Zhan, Bowen Hu, Xiaoguang Ren, Dapeng Tao, and Long Lan. Nt-fan: A simple yet effective noise-tolerant few-shot adaptation network. *Artificial Intelligence*, 346:104363, 2025.

[55] Haoang Chi, Feng Liu, Wenjing Yang, Long Lan, Tongliang Liu, Bo Han, Gang Niu, Mingyuan Zhou, and Masashi Sugiyama. Meta discovery: Learning to discover novel classes given very limited data. In *International Conference on Learning Representations*, 2022.

[56] Haoang Chi, Wenjing Yang, Feng Liu, Long Lan, Tao Qin, and Bo Han. Does confusion really hurt novel class discovery? *International Journal of Computer Vision*, 132(8):3191–3207, 2024.

[57] Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670, 2024.

[58] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[59] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

[60] Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. In *International conference on machine learning*, pages 10–32. PMLR, 2022.

[61] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[62] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[63] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

[64] Qi Wang, Marco Federici, and Herke van Hoof. Bridge the inference gaps of neural processes via expectation maximization. In *The Eleventh International Conference on Learning Representations*, 2022.

[65] Feng Xu, Shengyi Jiang, Hao Yin, Zongzhang Zhang, Yang Yu, Ming Li, Dong Li, and Wulong Liu. Enhancing context-based meta-reinforcement learning algorithms via an efficient task encoder (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15937–15938, 2021.

[66] Samuel Ritter, Jane Wang, Zeb Kurth-Nelson, Siddhant Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. Been there, done that: Meta-learning with episodic recall. In *International conference on machine learning*, pages 4354–4363. PMLR, 2018.

[67] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[68] Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021.

[69] Cevahir Koprulu, Thiago D Simão, Nils Jansen, and Ufuk Topcu. Risk-aware curriculum generation for heavy-tailed task distributions. In *Uncertainty in Artificial Intelligence*, pages 1132–1142. PMLR, 2023.

[70] Cheems Wang, Yiqin Lv, Yixiu Mao, Yun Qu, Yi Xu, and Xiangyang Ji. Robust fast adaptation from adversarially explicit task distribution generation. *arXiv preprint arXiv:2407.19523*, 2024.

[71] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[72] Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. RLVR without ineffective samples: Group prioritized off-policy optimization for llm reasoning. *arXiv preprint arXiv:2606.01281*, 2026.

[73] Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Yingyue Li, Wutong Xu, Lizhou Cai, Weijie Liu, Clive Bai, et al. Listwise policy optimization: Group-based rlvr as target-projection on the llm response simplex. *arXiv preprint arXiv:2605.06139*, 2026.

[74] Heming Zou, Qi Wang, Yun Qu, Yuhang Jiang, Lizhou Cai, Yixiu Mao, Ru Peng, Xin Xu, Weijie Liu, Kai Yang, et al. Trace: A unified rollout budget allocation framework for efficient agentic reinforcement learning. *arXiv preprint arXiv:2606.11119*, 2026.

[75] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

A Related Work

Risk-sensitive optimization for adaptive sequential decision-making. Risk considerations have long been central to sequential decision-making, where poor adaptation may lead to severe downstream failures [39, 40, 41, 42, 43]. This has motivated a broad line of work on incorporating risk-averse objectives into policy optimization [44, 45, 46, 47]. In randomized environments, including DR [48, 8] and Meta-RL [9, 49, 50], several optimization principles have been adopted to improve robustness. Empirical risk minimization (ERM) [30] optimizes the average loss under the training task distribution and is therefore commonly used as a standard baseline. Distributionally robust risk minimization (DRM) instead emphasizes high-risk tasks via conditional value-at-risk (CVaR) objectives [29, 17, 18, 32], while group distributionally robust optimization (GDRM) mitigates subpopulation shifts by reweighting group-level risks [31]. Worst-case meta-learning has also been investigated [51], but typically requires careful relaxations to maintain stable training. A shared limitation of these approaches is their reliance on extensive policy evaluation over large task spaces, which makes purely evaluation-driven robust optimization expensive in practice.

Robust active task sampling and risk-predictive models. Robust active task sampling (RATS) addresses the above efficiency bottleneck by replacing repeated exhaustive evaluation with a learned RPM [22]. Instead of executing the current policy on every candidate task, RATS estimates task risk through the RPM and then selects an informative or high-risk subset for policy optimization. MPTS [22] instantiates this idea by modeling the optimization trajectory as a generative sequence and using streaming variational inference to predict adaptation risk on unseen tasks. More recently, PDTS [23] reinterprets RATS from a task-selection perspective, formulates subset selection as an infinitely many-armed bandit problem, and combines posterior sampling with diversity regularization to alleviate concentration under large candidate batches. These methods substantially improve the efficiency of robust task selection; however, their predictive modules remain challenged by two factors: early-stage histories provide only sparse supervision, and risk estimates are inherently delayed because predictions are made before the next policy update is observed. Our method focuses on

these RPM-level limitations by introducing in-context sequence modeling and a one-step look-ahead prediction mechanism.

Cross-task adaptation and curriculum learning. Zero-shot and few-shot adaptation are two central evaluation regimes for adaptive decision-makers. DR aims to obtain zero-shot generalization by training over randomized environments [48, 8, 36, 52, 53, 54, 55, 56, 57], whereas meta-learning seeks reusable priors that enable rapid adaptation from limited experience [58]. Representative Meta-RL methods include optimization-based algorithms that learn initializations [9, 59, 60, 61], context-based approaches that infer latent task representations from support trajectories [62, 63, 64, 65], and recurrent architectures that internalize adaptation through memory [66]. Curriculum learning provides a complementary mechanism by changing which tasks are presented during training. For example, unsupervised environment design and its variants generate tasks with increasing challenge [67, 68, 69], while adversarial task-distribution methods construct curricula through learned generative distributions [70]. Our work is orthogonal to such curriculum-design strategies: rather than modifying the task distribution itself, we improve the accuracy and timeliness of risk estimation under a given distribution, making the approach potentially compatible with existing curricula.

Reinforcement learning for LLM post-training. Reinforcement learning from human feedback (RLHF) has become a standard approach for aligning large language models with human preferences [10, 71]. Recent post-training pipelines further use reinforcement learning to improve reasoning and task-solving capabilities [72, 73], where prompts can be viewed as task instances and responses are scored by reward models or verifiers [38]. Prompt-level curricula that emphasize informative or challenging examples have also been explored to improve training efficiency [14, 72, 74]. A practical obstacle is that estimating prompt difficulty usually requires costly rollouts from the current model. Our framework can be naturally mapped to this setting by treating prompts as task identifiers and using the RPM to estimate prompt-level difficulty, thereby supporting curriculum construction for LLM post-training without additional generation overhead.

B Expanded Proofs for Theoretical Analysis

This appendix contains the full proofs of all theoretical results stated in the main text. We first collect notation, then prove Theorem 3.1 and Proposition 3.2.

Notation. $\mathcal{T}$: task space; Θ: parameter space; $\boldsymbol{\theta}_t \in \Theta$: agent parameters at step t; $L(\boldsymbol{\theta}, \boldsymbol{\tau})$: true risk of task $\boldsymbol{\tau} \in \mathcal{T}$ under $\boldsymbol{\theta}$; $H_t = \{(\boldsymbol{\tau}_{t,i}, \ell_{t,i})\}_{i=1}^{B}$: per-iteration history with batch size B; $H_{0:t} = \bigcup_{k=0}^{t} H_k$: cumulative trajectory; $\mathcal{F}$: RPM hypothesis class; $\mathfrak{R}_n(\mathcal{F})$: Rademacher complexity with n samples. The vanilla RPM $\hat{\ell}_{\text{vanilla}}$ is trained on H_t (size B). ICRPM $\hat{\ell}_{\text{ICRPM}} = f_\phi(\boldsymbol{\tau}, H_{0:t})$ is trained on W consecutive batches (size $N = WB$).

Theorem 3.1 (Per-step Lagged Prediction Bound for Vanilla RPMs) Assume that $\ell(\boldsymbol{\tau}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\tau}) \in [0, M]$ for all $\boldsymbol{\theta} \in \Theta$, $\boldsymbol{\tau} \in \mathcal{T}$, and that $\ell(\boldsymbol{\tau})$ is L-Lipschitz in $\boldsymbol{\theta}$. The policy update uses a normalized average loss computed by i.i.d. samples and $\|\nabla_{\boldsymbol{\theta}} \ell_t(\boldsymbol{\tau}_i)\| \le G$. Let $\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau})$ denote the predictive risk function induced by the vanilla RPM at iteration t. Define $\epsilon_{\text{app}}^{\text{rec}} := \inf_{\boldsymbol{\psi},\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{\tau}\sim p}\big[\big(\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau}) - \ell_t(\boldsymbol{\tau})\big)^2\big]$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of H_t,

$$\mathbb{E}_{\boldsymbol{\tau}\sim p}\big[|\hat{\ell}_t(\boldsymbol{\tau}) - \ell_{t+1}(\boldsymbol{\tau})|\big] \le \underbrace{\sqrt{\epsilon_{\text{app}}^{\text{rec}} + \epsilon_{\text{gen}}^{\text{rec}}(B,\delta)}}_{\text{①Reconstruction Approximation Error}} + \underbrace{L\eta G}_{\text{②Temporal Lagging Error}}. \tag{6}$$

If the reconstruction loss class $\mathcal{G}^{\text{rec}} = \{(\boldsymbol{\tau}, y) \mapsto (\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau}) - y)^2 : \boldsymbol{\psi} \in \Psi, \boldsymbol{\phi} \in \Phi\}$ has Rademacher complexity $\mathfrak{R}_B(\mathcal{G}^{\text{rec}})$ and is bounded by M^2, then $\epsilon_{\text{gen}}^{\text{rec}}(B,\delta) = 4\mathfrak{R}_B(\mathcal{G}^{\text{rec}}) + 2M^2\sqrt{\frac{\ln(2/\delta)}{2B}}$.

Proof. Assume the RPM uses a fixed-variance Gaussian decoder $p_{\boldsymbol{\psi}}(\ell|\boldsymbol{\tau}, \boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\psi}}(\boldsymbol{\tau}, \boldsymbol{z}), \sigma^2)$, and define the predictive mean as $\hat{\ell}_{\boldsymbol{\psi},\boldsymbol{\phi},t}(\boldsymbol{\tau}) := \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|H_t)}[\mu_{\boldsymbol{\psi}}(\boldsymbol{\tau}, \boldsymbol{z}_t)]$. For such a decoder, the reconstruction term in the negative log-likelihood is proportional to squared reconstruction error:

$$-\log p_{\boldsymbol{\psi}}(\ell_t(\boldsymbol{\tau})|\boldsymbol{\tau}, \boldsymbol{z}_t) = \frac{1}{2\sigma^2}\big(\mu_{\boldsymbol{\psi}}(\boldsymbol{\tau}, \boldsymbol{z}_t) - \ell_t(\boldsymbol{\tau})\big)^2 + C_\sigma, \tag{7}$$

where C_σ is independent of (ψ, ϕ). Therefore, the reconstruction component of the ELBO in Eq. (2) induces the squared reconstruction objective analyzed below, while the KL term acts as a latent regularizer and is absorbed into the approximation and optimization error.

Let $f_{\psi,\phi,t}(\tau) := \hat{\ell}_{\psi,\phi,t}(\tau)$ and define the empirical and population squared reconstruction risks as

$$\widehat{\mathcal{R}}_t(f_{\psi,\phi,t}) := \frac{1}{B}\sum_{i=1}^{B}\big(f_{\psi,\phi,t}(\tau_{t,i}) - \ell_t(\tau_{t,i})\big)^2, \tag{8}$$

and

$$\mathcal{R}_t(f_{\psi,\phi,t}) := \mathbb{E}_{\tau\sim p}\big[\big(f_{\psi,\phi,t}(\tau) - \ell_t(\tau)\big)^2\big]. \tag{9}$$

By the standard Rademacher uniform convergence bound applied to $\mathcal{G}^{\mathrm{rec}}$, with probability at least $1-\delta$, uniformly over $(\psi, \phi) \in \Psi \times \Phi$,

$$\mathcal{R}_t(f_{\psi,\phi,t}) \le \widehat{\mathcal{R}}_t(f_{\psi,\phi,t}) + 2\mathfrak{R}_B(\mathcal{G}^{\mathrm{rec}}) + M^2\sqrt{\frac{\ln(2/\delta)}{2B}}. \tag{10}$$

Let $(\hat{\psi}, \hat{\phi})$ denote the parameters obtained by minimizing the empirical squared reconstruction objective, and let $f^*_{\psi,\phi,t}$ be a population minimizer in the class. Applying the above inequality to the empirical minimizer gives

$$\begin{aligned}\mathcal{R}_t(\hat{\ell}_t) &\le \widehat{\mathcal{R}}_t(\hat{\ell}_t) + 2\mathfrak{R}_B(\mathcal{G}^{\mathrm{rec}}) + M^2\sqrt{\frac{\ln(2/\delta)}{2B}} \\ &\le \widehat{\mathcal{R}}_t(f^*_{\psi,\phi,t}) + 2\mathfrak{R}_B(\mathcal{G}^{\mathrm{rec}}) + M^2\sqrt{\frac{\ln(2/\delta)}{2B}}.\end{aligned} \tag{11}$$

Applying the uniform convergence bound once more to $f^*_{\psi,\phi,t}$ yields

$$\widehat{\mathcal{R}}_t(f^*_{\psi,\phi,t}) \le \mathcal{R}_t(f^*_{\psi,\phi,t}) + 2\mathfrak{R}_B(\mathcal{G}^{\mathrm{rec}}) + M^2\sqrt{\frac{\ln(2/\delta)}{2B}}. \tag{12}$$

Combining the last two inequalities and using the definition of $\epsilon^{\mathrm{rec}}_{\mathrm{app}}$ gives

$$\begin{aligned}\mathcal{R}_t(\hat{\ell}_t) &\le \epsilon^{\mathrm{rec}}_{\mathrm{app}} + 4\mathfrak{R}_B(\mathcal{G}^{\mathrm{rec}}) + 2M^2\sqrt{\frac{\ln(2/\delta)}{2B}} \\ &= \epsilon^{\mathrm{rec}}_{\mathrm{app}} + \epsilon^{\mathrm{rec}}_{\mathrm{gen}}(B,\delta).\end{aligned} \tag{13}$$

Thus,

$$\mathbb{E}_{\tau\sim p}\big[(\hat{\ell}_t(\tau) - \ell_t(\tau))^2\big] \le \epsilon^{\mathrm{rec}}_{\mathrm{app}} + \epsilon^{\mathrm{rec}}_{\mathrm{gen}}(B,\delta). \tag{14}$$

It remains to account for the temporal shift from θ_t to θ_{t+1}. By the L_2 triangle inequality and Jensen's inequality,

$$\begin{aligned}&\mathbb{E}_{\tau\sim p}\big[|\hat{\ell}_t(\tau) - \ell_{t+1}(\tau)|\big] \\ &\le \big(\mathbb{E}_{\tau\sim p}[(\hat{\ell}_t(\tau) - \ell_{t+1}(\tau))^2]\big)^{1/2} \\ &\le \big(\mathbb{E}_{\tau\sim p}[(\hat{\ell}_t(\tau) - \ell_t(\tau))^2]\big)^{1/2} + \big(\mathbb{E}_{\tau\sim p}[(\ell_t(\tau) - \ell_{t+1}(\tau))^2]\big)^{1/2}.\end{aligned} \tag{15}$$

The first term is controlled by the reconstruction bound above. For the second term, since $\ell(\tau)$ is L-Lipschitz in θ,

$$|\ell_t(\tau) - \ell_{t+1}(\tau)| \le L\|\theta_{t+1} - \theta_t\|. \tag{16}$$

The normalized policy update and the gradient bound imply

$$\begin{aligned}\|\theta_{t+1} - \theta_t\| &= \left\|\eta\frac{1}{B}\sum_{i=1}^{B}\nabla_\theta \ell_t(\tau_i)\right\| \\ &\le \eta\frac{1}{B}\sum_{i=1}^{B}\|\nabla_\theta \ell_t(\tau_i)\| \le \eta G.\end{aligned} \tag{17}$$

Therefore,

$$\big(\mathbb{E}_{\tau\sim p}[(\ell_t(\tau) - \ell_{t+1}(\tau))^2]\big)^{1/2} \le L\eta G. \tag{18}$$

Substituting the two bounds gives

$$\mathbb{E}_{\tau\sim p}\big[|\hat{\ell}_t(\tau) - \ell_{t+1}(\tau)|\big] \le \sqrt{\epsilon^{\mathrm{rec}}_{\mathrm{app}} + \epsilon^{\mathrm{rec}}_{\mathrm{gen}}(B,\delta)} + L\eta G, \tag{19}$$

which completes the proof. □

B.1 Proof of Proposition 3.2

Proposition 3.2 (Look-ahead reduces temporal mismatch, restated). *Let $\boldsymbol{\Delta}_t = \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$ with $\|\boldsymbol{\Delta}_t\| \le \eta G$, and assume that $\ell(\boldsymbol{\theta}, \boldsymbol{\tau})$ is β-smooth in $\boldsymbol{\theta}$. For analysis, define the first-order look-ahead risk as $m_t(\boldsymbol{\tau}) = \ell_t(\boldsymbol{\tau}) + \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_t, \boldsymbol{\tau})^\top \boldsymbol{\Delta}_t$. If the ICRPM predictor satisfies $\mathbb{E}_{\boldsymbol{\tau}\sim p} |\hat{\ell}^{\mathrm{IC}}_{t+1}(\boldsymbol{\tau}) - m_t(\boldsymbol{\tau})| \le \epsilon_{\mathrm{IC}}$, then*

$$\mathbb{E}_{\boldsymbol{\tau}\sim p}\left[|\hat{\ell}^{\mathrm{IC}}_{t+1}(\boldsymbol{\tau}) - \ell_{t+1}(\boldsymbol{\tau})| \right] \le \epsilon_{\mathrm{IC}} + \frac{\beta}{2}\eta^2 G^2. \tag{20}$$

Proof. For any fixed task $\boldsymbol{\tau}$, denote $\ell_{t+1}(\boldsymbol{\tau}) = \ell(\boldsymbol{\theta}_{t+1}, \boldsymbol{\tau})$ and $\ell_t(\boldsymbol{\tau}) = \ell(\boldsymbol{\theta}_t, \boldsymbol{\tau})$. Since $\ell(\boldsymbol{\theta}, \boldsymbol{\tau})$ is β-smooth in $\boldsymbol{\theta}$, the standard first-order remainder bound gives

$$\left| \ell_{t+1}(\boldsymbol{\tau}) - \ell_t(\boldsymbol{\tau}) - \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}_t, \boldsymbol{\tau})^\top \boldsymbol{\Delta}_t \right| \le \frac{\beta}{2} \|\boldsymbol{\Delta}_t\|^2 \le \frac{\beta}{2}\eta^2 G^2. \tag{21}$$

By the definition of $m_t(\boldsymbol{\tau})$, this implies $|\ell_{t+1}(\boldsymbol{\tau}) - m_t(\boldsymbol{\tau})| \le \frac{\beta}{2}\eta^2 G^2$. Therefore, using the triangle inequality and taking expectation over $\boldsymbol{\tau} \sim p$,

$$\mathbb{E}_{\boldsymbol{\tau}\sim p}\left[|\hat{\ell}^{\mathrm{IC}}_{t+1}(\boldsymbol{\tau}) - \ell_{t+1}(\boldsymbol{\tau})| \right] \le \mathbb{E}_{\boldsymbol{\tau}\sim p}\left[|\hat{\ell}^{\mathrm{IC}}_{t+1}(\boldsymbol{\tau}) - m_t(\boldsymbol{\tau})| \right] + \frac{\beta}{2}\eta^2 G^2 \le \epsilon_{\mathrm{IC}} + \frac{\beta}{2}\eta^2 G^2. \tag{22}$$

This completes the proof. □

C Additional Preliminaries and Training Protocols

This appendix summarizes the task formulations used in our experiments, covering zero-shot domain randomization, few-shot meta-reinforcement learning, and RL post-training with verifiable rewards. Although these settings differ in their interaction modalities, they share a common structure: at each iteration, a subset of tasks or prompts is selected, evaluated by the current policy, and appended to the optimization history for subsequent predictive sampling.

C.1 Task Distributions in Randomized Environments

Definition C.1 (Identifier-induced MDP family)**.** Let $p(\boldsymbol{\tau})$ be a distribution over task identifiers. Each identifier $\boldsymbol{\tau}$ specifies an MDP $\mathcal{M}_{\boldsymbol{\tau}} = \langle \mathcal{S}, \mathcal{A}, P_{\boldsymbol{\tau}}, R_{\boldsymbol{\tau}}, \bar{\lambda} \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ are shared state and action spaces, while the transition kernel $P_{\boldsymbol{\tau}}(\boldsymbol{s}' \mid \boldsymbol{s}, \boldsymbol{a})$ and/or the reward function $R_{\boldsymbol{\tau}}(\boldsymbol{s}, \boldsymbol{a})$ may vary with $\boldsymbol{\tau}$. For an H-horizon trajectory $\xi = (\boldsymbol{s}_0, \boldsymbol{a}_0, r_0, \ldots, \boldsymbol{s}_H)$, the discounted return is

$$G_{\boldsymbol{\tau}}(\xi) = \sum_{h=0}^{H-1} \bar{\lambda}^h r_h. \tag{23}$$

This formulation covers continuous-control tasks in robotics, where $\boldsymbol{\tau}$ may encode physical parameters, target velocities, masses, lighting conditions, or goal positions. It also provides the task-level abstraction used by robust task sampling, where the sampler selects identifiers whose predicted difficulty is informative for policy improvement.

C.2 Zero-Shot and Few-Shot Adaptive Decision-Making

Domain randomization. In zero-shot domain randomization, the policy is trained over MDPs sampled from $p(\boldsymbol{\tau})$ and is directly evaluated on unseen environments without an adaptation phase. The standard objective is

$$\max_{\boldsymbol{\theta}} \ \mathcal{J}_{\mathrm{DR}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau})} \mathbb{E}_{\xi\sim\pi_{\boldsymbol{\theta}}, \mathcal{M}_{\boldsymbol{\tau}}} \left[\sum_{h=0}^{H-1} \bar{\lambda}^h r_h \right]. \tag{24}$$

For risk-based sampling, we use the negative return as the task risk,

$$\ell(\boldsymbol{\tau}; \boldsymbol{\theta}) = -\mathbb{E}_{\xi\sim\pi_{\boldsymbol{\theta}}, \mathcal{M}_{\boldsymbol{\tau}}} \left[\sum_{h=0}^{H-1} \bar{\lambda}^h r_h \right]. \tag{25}$$

In our experiments, TD3 is used for physical domain randomization [75].

Meta-RL with MAML. For few-shot adaptation, we use MAML [9] as the meta-learning backbone. Each task $\boldsymbol{\tau}$ provides support trajectories $\mathcal{D}_{\boldsymbol{\tau}}^S$ for fast adaptation and query trajectories $\mathcal{D}_{\boldsymbol{\tau}}^Q$ for post-adaptation evaluation. The inner update is

$$\boldsymbol{\theta}'_{\boldsymbol{\tau}} = \boldsymbol{\theta} - \eta_{\text{in}} \nabla_{\boldsymbol{\theta}} \ell(\mathcal{D}_{\boldsymbol{\tau}}^S; \boldsymbol{\theta}), \tag{26}$$

and the meta-objective is

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau})} \left[\ell(\mathcal{D}_{\boldsymbol{\tau}}^Q; \boldsymbol{\theta}'_{\boldsymbol{\tau}}) \right]. \tag{27}$$

Here, ℓ is instantiated as the negative query return after fast adaptation. Thus, DR can be viewed as the zero-shot case with no support set, while Meta-RL uses $\mathcal{D}_{\boldsymbol{\tau}}^S$ before evaluating $\mathcal{D}_{\boldsymbol{\tau}}^Q$.

C.3 RL Post-Training with Verifiable Rewards

We also consider RL post-training for LLM reasoning. Here, each identifier $\boldsymbol{\tau}$ denotes a prompt, or its embedding representation, sampled from a prompt pool $\mathcal{T}$. At iteration t, the policy $\pi_{\boldsymbol{\theta}_t}$ generates K responses $\boldsymbol{y}_t^{\boldsymbol{\tau}} = \{y_{t,i}^{\boldsymbol{\tau}}\}_{i=1}^K$ for each selected prompt. A verifier assigns rewards $\boldsymbol{r}_t^{\boldsymbol{\tau}} = \{r(\boldsymbol{\tau}, y_{t,i}^{\boldsymbol{\tau}})\}_{i=1}^K$. The latent success rate is

$$\rho_t^{\boldsymbol{\tau}} = \mathbb{E}[r_t^{\boldsymbol{\tau}}], \tag{28}$$

which measures the probability that the current policy solves prompt $\boldsymbol{\tau}$. The optimization history is written as

$$H_t = \left\{ \mathcal{T}_t^B, \mathcal{R}_t^B \right\}, \quad \mathcal{R}_t^B = \left\{ \boldsymbol{r}_t^{\boldsymbol{\tau}_{t,j}} \right\}_{j=1}^B,$$

where each $\boldsymbol{r}_i^{\boldsymbol{\tau}_{i,j}}$ denotes the reward vector over K responses for prompt $\boldsymbol{\tau}_{i,j}$, and B is the number of selected prompts in the batch. RLVR optimizes the expected verifiable reward:

$$\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{\tau} \sim \mathcal{T},\, y \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\tau})} \left[r(\boldsymbol{\tau}, y) \right]. \tag{29}$$

In practice, we use GRPO [38], which replaces the value network with a group-normalized advantage. For responses generated from the same prompt, the advantage is

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^K)}{\text{std}(\{r_i\}_{i=1}^K)}, \tag{30}$$

and the policy is updated with a clipped importance-ratio objective regularized by the KL divergence to a reference policy.

Online prompt selection. Prompt selection is needed because exact rollout-based evaluation is expensive. Evaluation-based methods over-sample a candidate pool $\mathcal{T}_t^{\hat{B}}$ and keep prompts with non-degenerate reward signals, for example those satisfying $\text{std}(\{r_i^{\boldsymbol{\tau}}\}_{i=1}^K) > 0$. Such filtering is effective but requires extra generations. Prediction-based methods reduce this cost by estimating prompt difficulty, typically through a predictive model of $p(\rho_t^{\boldsymbol{\tau}} \mid \boldsymbol{\tau}, \boldsymbol{\theta}_t)$ or its history-conditioned variant. However, prompt-wise independent estimators may fail to share information across prompts and may lag behind the changing policy.

C.4 Unified Predictive Selection View

Both robust task sampling and active prompt selection can be expressed as selection from predicted outcomes. Let $\hat{y}_{\boldsymbol{\tau}}^t$ denote the model-predicted outcome for candidate $\boldsymbol{\tau}$. For active prompt selection, this outcome is the predicted success rate, $\hat{y}_{\boldsymbol{\tau}}^t = \hat{\rho}_{\boldsymbol{\tau}}^t$, and the target is an intermediate difficulty level ρ^*. For robust task sampling, the outcome is the predicted risk, $\hat{y}_{\boldsymbol{\tau}}^t = \hat{\ell}_t(\boldsymbol{\tau})$, and the target is the largest predicted risk in the candidate pool. The selected batch is

$$\mathcal{T}_t^B = \arg\text{top-}B_{\boldsymbol{\tau} \in \mathcal{T}_t^{\hat{B}}} \left(-\left\| \hat{y}_{\boldsymbol{\tau}}^t - c_t^* \right\|_2^2 \right), \tag{31}$$

where $c_t^* = \rho^*$ for prompt selection and $c_t^* = \max_{\boldsymbol{\tau} \in \mathcal{T}_t^{\hat{B}}} \hat{\ell}_t(\boldsymbol{\tau})$ for robust task sampling.

D Risk-Aware Evaluation and Predictive Ranking Metrics

This section summarizes the metrics and regularity conditions used to evaluate tail-task robustness and risk-prediction quality. We write the task risk induced by a model parameterized by $\boldsymbol{\theta}$ as

$$\ell_{\boldsymbol{\theta}}(\boldsymbol{\tau}) := \ell(\mathcal{D}^Q_{\boldsymbol{\tau}}, \mathcal{D}^S_{\boldsymbol{\tau}}; \boldsymbol{\theta}),$$

where $\boldsymbol{\tau} \sim p(\boldsymbol{\tau})$ is a task identifier. Depending on the benchmark, $\ell_{\boldsymbol{\theta}}(\boldsymbol{\tau})$ is instantiated as MSE for regression, negative return for control, or failure probability for RL post-training.

D.1 CVaR_α for Tail-Task Robustness

Definition D.1 (CVaR_α)**.** Let $F_{\boldsymbol{\theta}}(z) = \Pr_{\boldsymbol{\tau}\sim p(\boldsymbol{\tau})}\left(\ell_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \le z\right)$ be the cumulative distribution function of the task risk, and define its α-quantile as $q_\alpha(\boldsymbol{\theta}) = \inf\{z : F_{\boldsymbol{\theta}}(z) \ge \alpha\}$. Following [29], the α-level Conditional Value-at-Risk is the expected risk over the upper tail:

$$\mathrm{CVaR}_\alpha(\boldsymbol{\theta}) = \mathbb{E}\left[\ell_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \mid \ell_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \ge q_\alpha(\boldsymbol{\theta})\right]. \tag{32}$$

Equivalently, for a continuous risk distribution,

$$\mathrm{CVaR}_\alpha(\boldsymbol{\theta}) = \int z\, dF^\alpha_{\boldsymbol{\theta}}(z), \qquad F^\alpha_{\boldsymbol{\theta}}(z) = \begin{cases} 0, & z < q_\alpha(\boldsymbol{\theta}), \\ \dfrac{F_{\boldsymbol{\theta}}(z) - \alpha}{1-\alpha}, & z \ge q_\alpha(\boldsymbol{\theta}), \end{cases} \tag{33}$$

where $F^\alpha_{\boldsymbol{\theta}}$ is the normalized tail-risk distribution. We denote the corresponding tail-task density by $p_\alpha(\boldsymbol{\tau}; \boldsymbol{\theta})$.

Under this convention, $\mathrm{CVaR}_{0.9}$ evaluates the expected risk on the hardest 10% of tasks. Thus, lower CVaR indicates stronger tail-task robustness. For control tasks, where risk is defined as negative return, reporting $-\mathrm{CVaR}_\alpha$ gives the corresponding CVaR return, for which higher values are better.

D.2 Spearman Rank Correlation for Risk Prediction

While CVaR_α measures the robustness of the learned policy, it does not directly assess whether the risk-predictive model correctly identifies hard tasks. Since predictive samplers mainly rely on the relative ordering of candidate tasks, we additionally report Spearman Rank Correlation (SRC) between predicted and observed risks.

Given a candidate or validation set $\{\boldsymbol{\tau}_i\}_{i=1}^N$, let $\hat{\ell}_i = \hat{\ell}(\boldsymbol{\tau}_i)$ be the predicted risk and $\ell_i = \ell(\mathcal{D}^Q_{\boldsymbol{\tau}_i}, \mathcal{D}^S_{\boldsymbol{\tau}_i}; \boldsymbol{\theta})$ be the observed risk. Denoting the rank operator by $\mathrm{rk}(\cdot)$, SRC is computed as

$$\mathrm{SRC} = 1 - \frac{6\sum_{i=1}^N \left(\mathrm{rk}(\hat{\ell}_i) - \mathrm{rk}(\ell_i)\right)^2}{N(N^2-1)}. \tag{34}$$

SRC lies in $[-1, 1]$. Values close to 1 indicate that the predicted risk order matches the observed task-difficulty order. This metric is particularly relevant for predictive task sampling: even if the absolute risk values are not perfectly calibrated, a high SRC implies that the sampler can still prioritize informative or high-risk tasks. In contrast, low SRC indicates rank-flipping in the predicted difficulty landscape, which may lead to suboptimal task selection and weaker tail-task performance.

E Experiment Details

For all baselines, we follow the default configurations reported in their original papers whenever available, including model architectures, optimizer settings, task-selection rules, and hyperparameters. When a baseline is adapted to our shared MPTS-style evaluation protocol, we keep the training budget, candidate-pool size, batch size, number of seeds, and evaluation schedule consistent with ICRPM for fair comparison.

E.1 Meta-Reinforcement Learning Experiments

Shared Setup We evaluate on three MuJoCo Meta-RL benchmarks (**HalfCheetahVel**, **Walker2dVel**, **ReacherPos**) from the MAML task-distribution suite. Tasks are parameterised by continuous descriptors $\boldsymbol{\tau}$ (target velocity or goal position) sampled from $p(\boldsymbol{\tau})$ at each iteration. All methods share the same meta-learner: **MAML** with TRPO as the inner optimiser. Key shared hyperparameters are listed in Table 3.

Table 3: Shared meta-learning hyperparameters (Meta-RL).

Hyperparameter	Value
Meta-training iterations	500
Meta-batch size B	20 tasks
Validation batch size	20 tasks (uniform)
Inner adaptation steps	1
Fast batch size (trajectories / task)	20
Max KL divergence	0.01
Discount factor	0.99
GAE λ	1.0
Random seeds	10

Evaluation protocol. All methods are evaluated on a held-out validation set of 20 tasks sampled uniformly from $p(\boldsymbol{\tau})$ at each iteration; validation episodes are excluded from the environment interaction count.

ICRPM Configuration for Meta-RL ICRPM uses Transformer backbone RPM that learns a predictive distribution $p(\ell|\boldsymbol{\tau}, C_t)$ conditioned on the latest context set $C_t = \{(\boldsymbol{\tau}_{t,i}, \ell_t(\boldsymbol{\tau}_{t,i}))\}_{i=1}^{B}$.

Architecture Details. The encoder maps context pairs $\{(\boldsymbol{\tau}_i, \ell_i)\}_{i\in C_t}$ through a depth-4 MLP embedding, a Transformer encoder (self-attention over context), and a cross-attention module where each target task $\boldsymbol{\tau}$ attends to the encoded context. The decoder outputs Gaussian parameters (μ, σ) with $\sigma = 0.1 + 0.9 \cdot \mathrm{softplus}(\cdot)$. The transformer-based ICRPM is parameter-matched to the vanilla RPM for a fair comparison ($\approx$0.65 M parameters).

Training objective. At each iteration t, one gradient step is taken on the weighted NLL objective (Eq. (4) in the main text):

$$\min_{\boldsymbol{\phi}} \ \mathcal{J}(\boldsymbol{\phi}) = -\sum_{k=1}^{W} \gamma^k \left(\frac{1}{B} \sum_{i=1}^{B} \ln p_{\boldsymbol{\phi}}(\ell_{t-k+1}(\boldsymbol{\tau}_{t-k+1,i}) | \boldsymbol{\tau}_{t-k+1,i},\ H_{t-k}) \right), \tag{35}$$

where $H_{t-k} = \{(\boldsymbol{\tau}_{t-k,i}, \ell_{t-k,i})\}_{i=1}^{B}$ is the per-step history at time $t-k$, $W = 3$ is the sliding-window size, $\gamma = 0.9$ is the temporal decay, and Adam (lr $= 5 \times 10^{-3}$) is used.

Task selection. This retains the predictive selection setup in MPTS [22]. At each iteration, $1.5B$ candidate tasks are drawn from $p(\tau)$. ICRPM predicts the mean risk $\hat{y}_i = \mathbb{E}_{p_\phi(y|\hat{\tau}_i, H_t)}[y]$ for each candidate. For robust task sampling, we set $c^* = \max_i \hat{y}_i$ and select the B candidates with the largest score $-\|\hat{y}_i - c^*\|_2^2$, which is equivalent to selecting the top-B candidates by predicted risk in the scalar-risk case.

E.2 RLVR Countdown Experiment

Setup We use the **Countdown** mathematical reasoning task (20k training instances, Countdown-3to4) with **Qwen3-4B-Base** as the policy backbone $\pi_{\boldsymbol{\theta}}$, trained via **GRPO+** inside the **veRL** framework (v0.3.1) on 4×A100-80 GB GPUs. Key training hyperparameters are listed in Table 4.

The reward function follows the format-aware Countdown setting: $r = 1$ (correct), $r = 0.1$ (incorrect but well-formatted), $r = 0$ (otherwise). Evaluation reports pass@1 accuracy on held-out CD34 and the harder CD4 benchmark.

Table 4: RLVR training hyperparameters.

Hyperparameter	Value
Base model	Qwen3-4B-Base
Training steps	100
Prompt batch size B	128 (4-GPU)
Responses per prompt K	8
Max response length	1,024 tokens
Policy optimiser	AdamW, lr 1×10^{-6}
Entropy coefficient	0.001
KL coefficient	0.0
PPO clip ratio (low / high)	0.2 / 0.28

ICRPM Configuration for RLVR ICRPM replaces the prompt-difficulty predictor (RPM). At each training step, rollouts are generated by vLLM using the current actor. After reward computation, prompt embeddings $\boldsymbol{\tau} \in \mathbb{R}^{1024}$ are extracted from the current actor's last hidden states and paired with observed solve rates to form the context H_t, where the solve rate is the fraction of generated responses that exactly solve the Countdown instance. These task representations are not frozen; their policy-induced drift is treated as part of the non-stationary task-risk context modeled by ICRPM. The ICRPM is trained with the objective in Eq. (35), using $W = 3$, $\gamma = 0.9$, and 20 gradient steps per training step, matching the vanilla RPM. In RLVR, ICRPM is scaled to match the vanilla RPM parameter count (≈42.8 M). The task-sampling ratio is 80%.

E.3 Domain Randomization Experiments

For zero-shot DR we follow the MPTS protocol with **TD3** as the policy backbone on **Ergo-Reacher** and **Lunar-Lander** [36]. Each task $\boldsymbol{\tau}$ is an environment instance induced by a continuous domain-randomisation identifier; the risk $\ell(\boldsymbol{\tau}) = L(\boldsymbol{\theta}_t, \boldsymbol{\tau})$ is the negative episodic return without inner-loop adaptation. The same task-batch size B, candidate-pool size $\hat{B}$, and baseline protocol as Meta-RL are used. ICRPM uses a parameter-matched Transformer architecture and the same temporal augmentation settings ($W = 3, \gamma = 0.9$).

E.4 Few-Shot Sinusoid Regression

Each task is specified by $\boldsymbol{\tau} = (a, b)$ with $f_{\boldsymbol{\tau}}(x) = a \sin(x - b)$, $a \in [0.1, 5.0]$, $b \in [0, \pi]$. The meta-learner is MAML and task risk is query-set MSE after adaptation. We report average MSE and CVaR_α (Eq. (33)) with $\alpha \in \{0.5, 0.7, 0.9\}$, along with SRC (Eq. (34)) throughout training. ICRPM is parameter-matched to the vanilla RPM and uses the same temporal augmentation settings ($W = 3$, $\gamma = 0.9$).

E.5 Ablation Studies

Ablations are conducted on **Lunar-Lander DR** with 10 independent runs each. The ICRPM parameter count is equal to the vanilla RPM across all ablation studies for fair comparison. We vary four design dimensions: acquisition principle, RPM backbone, history augmentation, and the window/decay hyperparameters.

1. **Acquisition principle.** We compare worst-case selection by predicted risk $\mu(\ell)$ with a diversity-aware variant that combines $\mu(\ell)$ with a set-level diversity bonus. Both variants are evaluated under identical optimisation budgets and compared with PDTS [23]. This ablation tests whether the improvement comes from the risk predictor itself rather than a specific acquisition heuristic.
2. **RPM backbone and history augmentation.** We compare the default Transformer backbone with a parameter-matched MLP that retains the same context-conditioned look-ahead objective in Eq. (35). To isolate the role of history reuse from look-ahead prediction, we also evaluate *MPTS-augment*, which augments vanilla MPTS [22] with the same three-step history window and temporal decay $\gamma = 0.9$ used by ICRPM, but without the context-

conditioned one-step extrapolative objective. This comparison separates the effect of the backbone, the sliding-window augmentation, and the proposed look-ahead formulation.

3. **Look-back window W.** We vary $W \in \{1, 2, 3, 8\}$ with $\gamma = 0.9$ fixed. Smaller windows use fresher histories but provide less task coverage, while larger windows improve sample reuse but can introduce stale context from earlier policy states. We set $W = 3$ as the default trade-off between coverage and freshness.
4. **Temporal decay factor γ.** We vary $\gamma \in \{0.5, 0.7, 0.9, 1.0\}$ with the default window size fixed. Smaller values emphasize only the most recent batches, whereas $\gamma = 1.0$ weights stale and fresh histories equally. Among the tested values, $\gamma = 0.9$ provides a stable balance by mildly discounting older histories, and we therefore use it as the default.

All ablation settings report $\text{CVaR}_{0.9}$ return, average return, and SRC (Eq. (34)) on the task-distribution validation set. The results show that ICRPM remains effective across acquisition rules and backbone choices, while MPTS-augment does not match ICRPM under the same history-window and decay configuration. This indicates that the gains are not explained by history reuse alone, but mainly come from context-conditioned one-step look-ahead risk prediction.

E.6 Additional Experimental Results

We report additional diagnostics of next-step risk prediction, a controlled look-ahead/history-reuse ablation, and independent-run replication of the Countdown result. Unless stated otherwise, values are means $\pm$ standard errors of the mean (SEM) across independent training runs, with $\text{SEM} = s/\sqrt{n}$ for sample standard deviation s and number of runs n. The run counts are specified separately for each experiment.

E.6.1 Full-Trajectory Next-Step Prediction on Sinusoid

We evaluated next-step prediction throughout training to examine whether the look-ahead formulation improves risk tracking beyond selected checkpoints. The Sinusoid benchmark retains task-dependent risk drift while allowing evaluation at every epoch. Following the task definition in Appendix E.4, the diagnostic used a downstream learning rate of 5×10^{-4}, 20,000 epochs, 32 candidate tasks, and 16 metatasks, with five independent runs. ICRPM and MPTS were evaluated on the same candidate tasks at each epoch. The reported quantity is absolute next-step prediction error, averaged across candidates and the full training trajectory within each run, then summarized across runs.

Table 5: Full-trajectory next-step risk-prediction error on Sinusoid (five runs, mean $\pm$ SEM). Lower is better.

Method	Runs	Mean absolute next-step prediction error
MPTS	5	0.0728 ± 0.0036
ICRPM	5	$\mathbf{0.0593 \pm 0.0014}$

ICRPM reduced the mean absolute next-step prediction error by 18.5% relative to MPTS (Table 5). This diagnostic measures prediction error directly, rather than downstream regression MSE. Across the five runs, ICRPM had lower candidate-averaged error than MPTS in $65.67\% \pm 6.77\%$ of epochs (mean $\pm$ SEM of the per-run proportions). It provides empirical support for improved tracking in this setting, but does not establish that the sufficient condition in Proposition 3.2 holds at every epoch: the diagnostic does not separately estimate ϵ_{IC}, L, β, or G.

E.6.2 Sensitivity to Larger Downstream Updates

To examine sensitivity to the scale of adjacent updates, we multiplied the downstream model-update learning rate by 1, 2, 4, and 8, holding the remaining setup fixed. Each setting used five independent runs, two layers, $W = 3$, and 20,000 epochs. The changed learning rate belongs to the downstream learner, rather than the RPM optimizer. Table 6 reports downstream regression MSE and tail MSE, together with the SRC of predicted and observed task risk. These MSE quantities differ from the prediction-error diagnostic in Table 5.

Table 6: Sinusoid learning-rate stress test (five runs per setting, mean ± SEM). Lower MSE and higher SRC are better.

LR scale	Learning rate	Best MSE	Final MSE	$\mathrm{CVaR}_{0.9}$ MSE	SRC
$1\times$	0.0005	0.2110 ± 0.0163	0.2169 ± 0.0165	0.7637 ± 0.0750	0.7465 ± 0.0117
$2\times$	0.001	0.2648 ± 0.0598	0.2776 ± 0.0653	0.9800 ± 0.1932	0.7378 ± 0.0079
$4\times$	0.002	0.2141 ± 0.0157	0.2431 ± 0.0060	0.8646 ± 0.0230	0.6289 ± 0.0243
$8\times$	0.004	0.3095 ± 0.0354	0.3285 ± 0.0310	1.0651 ± 0.1096	0.5364 ± 0.0312

Every tested learning rate above the default had higher mean final MSE and tail MSE, and lower mean SRC. At $8\times$, SRC decreased from 0.7465 to 0.5364, while final MSE increased from 0.2169 to 0.3285. The MSE changes were not monotonic across all intermediate settings. These observations are consistent with reduced tracking quality as local predictability weakens. However, changing the learning rate also affects downstream optimization; the test does not isolate prediction error as the sole cause of the MSE changes. It tests sensitivity to increased update rates, rather than abrupt distribution shifts or injected reward noise. Robustness to those perturbations remains untested here.

E.6.3 Separating Look-Ahead Prediction from History Reuse

We extended the Lunar-Lander DR ablation in Appendix E.5 to quantify the contributions of context-conditioned look-ahead and multi-batch history reuse. All four variants used the shared learner and evaluation protocol, with ten independent runs each. MPTS used neither component. MPTS-augment reused the same $W = 3$ history window and temporal decay $\gamma = 0.9$ as full ICRPM, but retained the vanilla prediction objective. ICRPM with $W = 1$ retained the context needed for one-step prediction without reusing multiple historical transitions. Full ICRPM combined look-ahead with the $W = 3$ weighted history window.

Table 7: Component ablation on Lunar-Lander DR (ten runs, mean ± SEM). Higher returns are better. The history-reuse column denotes reuse of multiple historical transitions; $W = 1$ still retains context for look-ahead prediction.

Method	Look-ahead	History reuse	Uniform return	$\mathrm{CVaR}_{0.9}$ return
MPTS	No	No	281.6 ± 3.3	251.8 ± 4.1
MPTS-augment	No	Yes	275.6 ± 4.5	236.2 ± 2.4
ICRPM ($W = 1$)	Yes	No	288.8 ± 3.8	250.7 ± 4.1
ICRPM ($W = 3$)	Yes	Yes	$\mathbf{298.1 \pm 3.0}$	$\mathbf{265.8 \pm 2.1}$

History reuse alone did not explain the improvement: MPTS-augment had lower mean returns than MPTS on both metrics (Table 7). ICRPM with $W = 1$ improved mean Uniform return by 7.2 relative to MPTS, but did not improve mean $\mathrm{CVaR}_{0.9}$ return (250.7 versus 251.8). Combining look-ahead with weighted history reuse increased mean Uniform and $\mathrm{CVaR}_{0.9}$ returns over $W = 1$ by 9.3 and 15.1, respectively. Thus, the strongest observed performance required the combined design. This comparison does not isolate temporal weighting from history-window length; the separate window and decay studies are described in Appendix E.5.

E.6.4 Independent-Run Replication on Countdown

We completed two additional independent training runs for both Uniform and ICRPM, giving three runs per method including the runs reported in Table 2. The replication retained Qwen3-4B-Base and the same 100-step GRPO/veRL training protocol (Appendix E.2). Each evaluated checkpoint used 512 held-out CD34 prompts and 512 held-out CD4 prompts, with 16 sampled responses per prompt and deterministic verification, yielding 16,384 scored generations. These 16 evaluation responses are distinct from the $K = 8$ responses per prompt used during training.

ICRPM achieved a replicated mean advantage of 2.9 percentage points over Uniform (Table 8). The aggregate means differ from the individual-run values in Table 2, which reports 62.5% and 65.4%, respectively. The statistical unit for SEM is an independent training run, rather than a prompt or generated response. With only three runs per method, these summaries describe run-to-run variability; they are not formal confidence intervals or significance tests. This replication covers Uniform and

Table 8: Replicated endpoint accuracy on Countdown. Avg. is the average of CD4 and CD34 accuracy, following Table 2. Values are mean ± SEM across three independent training runs.

Method	Independent runs	Avg. accuracy (%)
Uniform	3	62.4 ± 0.1
ICRPM	3	$\mathbf{65.3 \pm 0.2}$

ICRPM only and does not establish a statistically significant advantage over the other baselines. The LLM-specific conclusion remains limited to Qwen3-4B-Base on Countdown.

E.7 Compute resources.

Table 9 reports approximate single-run wall-clock times, where each entry corresponds to one method on one benchmark/environment with one random seed, including training and periodic evaluation but excluding preliminary debugging runs. For Sinusoid and continuous-control experiments, ERM [30], MPTS [22], GDRM [31], OHTM [33], and ICRPM have comparable runtime, while DRM is about 30–40% slower due to its additional importance-weight computation. For RLVR Countdown, Uniform, PCL [14], MoPPS [13], and ICRPM have comparable runtime, while GRESO [34] is about 60% slower due to additional rollout-selection overhead.

Table 9: Approximate single-run compute resources.

Experiment	Hardware	Common	Slower
Sinusoid	RTX 4090 24GB	∼12.2 min	DRM: ∼16.4 min
DR	RTX 4090 24GB	∼3.0 h	DRM: ∼4.2 h
Meta-RL	A10 24GB	∼8.7 h	DRM: ∼11.6 h
RLVR	4×A100 80GB	∼11.1 h	GRESO: ∼17.8 h

E.8 Assets, Licenses, and Terms of Use.

All experiments use publicly available research assets. For LLM post-training, we use Qwen3-4B-Base as the policy backbone, and implement training and inference with veRL and vLLM. These model and software assets are released under the Apache License 2.0. For continuous-control experiments, we use OpenAI Gym with MuJoCo-based environments through the legacy `mujoco-py` interface. OpenAI Gym and `mujoco-py` are released under permissive open-source licenses, while the MuJoCo simulator is used subject to its applicable license terms; we do not redistribute MuJoCo binaries, license keys, or third-party assets. Our TD3 and MAML implementations are adapted from or benchmarked against publicly available research codebases released under permissive licenses. For Countdown, we use public arithmetic-reasoning datasets from Hugging Face for research evaluation only, do not redistribute raw dataset files, and provide sufficient preprocessing and evaluation details to support reproducibility. We comply with the terms of use of all upstream assets and retain the corresponding license and copyright notices where applicable.

F Limitations

ICRPM is proposed as a context-conditioned extrapolative RPM for improving risk approximation in MPTS-style task sampling. Several aspects remain open for future investigation.

1. **History-based extrapolation.** ICRPM relies on recent optimization histories to infer next-step task risk. This is effective when recent histories capture local risk drift, but may be less reliable under abrupt policy updates, highly noisy rewards, or task regions poorly covered by the history buffer.
2. **Scope of validation and analysis.** Our theory uses standard boundedness and smoothness assumptions to characterize temporal mismatch, and our experiments cover regression, robust continuous control, and RLVR-based LLM post-training. Extending the analysis

and evaluation to larger models, longer horizons, and broader task distributions remains an important direction for future work.

G Broader Impacts

ICRPM improves sample-efficient task selection by reducing temporal-lag bias in online risk prediction. This can lower interaction and compute costs and improve robustness on difficult or tail tasks in simulated control and verifiable-reward reasoning settings. However, more efficient task selection may also reduce the cost of RL-based post-training for capable models, including applications with potential misuse. Moreover, learned risk predictors may amplify blind spots if the task distribution is biased or incomplete. We therefore recommend using such methods with broad task coverage, safety-oriented stress tests, and task-distribution audits before any real-world deployment. Our experiments use simulated environments and algorithmic reasoning benchmarks only, and do not involve human subjects, private user data, or real-world deployment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe temporal lag and sparse-history risk prediction in MPTS, and the proposed context-conditioned look-ahead model. The theoretical claims are conditional on the stated assumptions, and the empirical claims are supported by the regression, continuous-control, and Countdown experiments in Section 4 and Appendix E.6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix F discusses dependence on informative recent histories, the boundedness and smoothness assumptions, and the limited empirical scope. Appendix E.6.2 reports sensitivity to larger downstream updates and distinguishes this test from untested abrupt shifts and reward noise.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 3.1 and Proposition 3.2 state the assumptions used in the analysis, including boundedness, smoothness, and approximation conditions. Full proofs are provided in Appendix B; the look-ahead result is conditional on the in-context approximation error.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method description specifies the predictor, training objective, and task-selection procedure, while Appendix E reports benchmark definitions, model configurations, hyperparameters, run counts, and evaluation protocols. Appendix E.6 provides the settings for the additional diagnostics, component ablation, and Countdown replication.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: A complete executable code package for all experiments is not provided with this paper. The algorithm and experimental settings are described in the main text and Appendix E.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix E describes the task distributions, learners, optimizers, principal hyperparameters, and evaluation settings for each benchmark. Appendix E.6 specifies the changed learning rates, ablation variants, and independent-run replication protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Unless stated otherwise, results are reported as mean $\pm$ SEM across independent training runs, with SEM $= s/\sqrt{n}$, where s is the sample standard deviation and n is the run count: $n = 10$ for the main Sinusoid and continuous-control experiments and the component ablation, $n = 5$ for the additional Sinusoid diagnostics, and $n = 3$ for the replicated Uniform/ICRPM Countdown comparison (Appendix E.6); Table 2 reports single-run results. The uncertainty reflects stochastic initialization, task sampling, and training, and SEM is a descriptive measure rather than a formal confidence interval or significance test.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E.7 reports GPU types, memory, and approximate per-run wall-clock times, including training and periodic evaluation. These timings refer to one method, benchmark, and training run, and exclude preliminary debugging runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The experiments use simulated control and verifiable-reward reasoning benchmarks without human-subject data collection, private user data, or real-world deployment. To the best of our knowledge, the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix G discusses potential reductions in interaction and compute costs, improved tail-task robustness, and risks from cheaper model post-training and biased task distributions. It also identifies task coverage, stress testing, and distribution audits as considerations before real-world deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release high-risk pretrained models or scraped datasets requiring controlled-access safeguards. An existing language model is fine-tuned only to evaluate the task-sampling method on Countdown.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix E.8 identifies the main third-party models, software, simulators, datasets, and codebases, and describes their licenses or applicable usage terms. Restricted simulator binaries and raw third-party datasets are not redistributed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work does not introduce or release a new dataset, pretrained model, or benchmark asset requiring accompanying asset documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The experiments use simulated environments and programmatically verifiable reasoning tasks. They do not involve crowdsourcing, human participants, or human-subject data collection.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human participants or human-subject data collection, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an original or non-standard component of the proposed risk-prediction methodology. Qwen3-4B-Base is used as the policy backbone in the RLVR evaluation, with its role and training protocol described in Section 4.3 and Appendix E.2.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.