TOWARDS UNPREDICTABLE WORLDS: CONTINUAL IN-CONTEXT REINFORCEMENT LEARNING IN NON-STATIONARY ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Traditional In-Context Reinforcement Learning (ICRL) demonstrates impressive rapid adaptation, but its reliance on static environments limits its applicability. In contrast, real-world scenarios are inherently non-stationary, with continuous and unpredictable changes that challenge an agent's ability to adapt. To bridge this gap, we formally define and systematically investigate Continual In-Context Reinforcement Learning in Non-Stationary Environments. Our central question is: what model architectures and training strategies enable an agent not only to rapidly master new dynamics in a continuously evolving environment, but also to efficiently discard or isolate outdated information, thereby achieving robust online adaptation? To ground our investigation, we construct a new benchmark suite featuring two complementary non-stationary domains—a symbolic reasoning task and a physics-based control task—each modified to exhibit unpredictable, intra-lifetime dynamic changes. On these benchmarks, we conduct extensive evaluations at both the model and training-strategy levels. At the model level, we compare state-of-the-art sequence model architectures. At the training strategy level, we systematically analyze the influence of stationary versus non-stationary training, dynamic change frequency, context length, and interaction scale. Our findings demonstrate the necessity of non-stationary training and reveal critical factors shaping continual adaptation. These results provide actionable insights and design principles for building agents capable of learning and adapting in truly open and dynamic worlds.

1 INTRODUCTION

Reinforcement Learning (RL) provides a powerful paradigm for solving sequential decision-making problems through trial-and-error, and has achieved significant progress across diverse domains (Sutton et al., 1998). Recently, with the advent of powerful sequence models such as the Transformer (Vaswani et al., 2017) and its variants, an emerging approach known as In-Context Reinforcement Learning (ICRL) has shown immense potential (Laskin et al., 2022; Lee et al., 2023; Grigsby et al., 2023; Moeini et al., 2025). ICRL enables pretrained sequence models to emulate RL-like adaptation during a forward pass (Lin et al., 2023), purely by processing contextual information, such as historical reward-observation-action sequences, without updating network parameters. This gradient-free, test-time adaptation not only improves computational efficiency but also offers a promising step toward general-purpose agents capable of rapid multi-task adaptation.

However, prior ICRL research (Laskin et al., 2022; Grigsby et al., 2023; Team et al., 2023) generally assumes that while agents must adapt to different task instances, the underlying dynamics of any given instance remain fixed throughout its lifetime. In other words, during ICRL's adaptation phase, the "rules" of the environment are static. This stands in stark contrast to real-world scenarios, which are often non-stationary and unpredictable. Examples include gradual drifts in physical parameters, such as robotic component wear, and abrupt shifts in task rules or objectives, such as updates to game mechanics (Hamadanian et al., 2023). Such non-stationarity poses a critical challenge: agents must not only learn new dynamics but also avoid inference from outdated context, a problem akin to catastrophic forgetting, but occuring within a single lifetime.

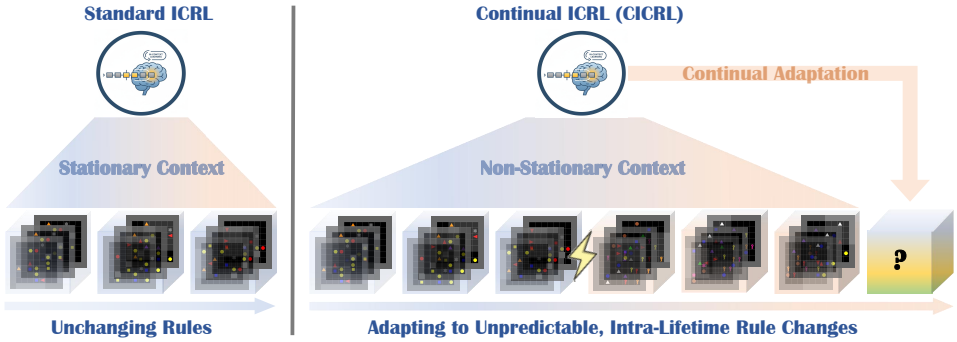

Figure 1: Comparison of Standard ICRL and Continual ICRL (CICRL). Standard ICRL adapts to fixed rules in a stationary environment, while CICRL requires an agent to continually adapt to unpredictable rule changes within a single lifetime.

Motivated by this gap, we explore the capabilities and limitations of ICRL in continuously non-stationary environments. We propose and investigate a central question: *when environmental dynamics shift within a single interaction sequence, can sequence models both rapidly adapt to new dynamics and effectively mitigate interference from obsolete context?* We refer to this extended setting as **Continual In-Context Reinforcement Learning (CICRL)**. As illustrated in Figure 1, unlike standard ICRL which adapts within a static environment with fixed rules, CICRL requires the agent to continuously adapt to unpredictable dynamic changes that occur within a single lifetime. CICRL requires agents to integrate the fast contextual adaptation strengths of ICRL with continual learning mechanisms that address intra-lifetime drift, thus bridging two previously separate lines of research.

To systematically investigate this problem, we first construct a new suite of non-stationary environment benchmarks. Specifically, we modify the rule-based XLand-Minigrid (Nikulin et al., 2024) grid-world suite and the physics-based Kinetix continuous control suite (Matthews et al., 2024b), introducing dynamic changes that occur within a single trajectory. These benchmarks compel agents to adapt online without relying on environment resets to clear its history. This intra-lifetime dynamic change provides a controllable platform for analyzing how sequence models manage context in the presence of evolving dynamics.

Using these benchmarks, we conduct comprehensive experimental evaluations and in-depth analyses of various advanced sequence model architectures on CICRL tasks. Alongside the widely-used Transformer (Vaswani et al., 2017), we evaluate emerging architectures that demonstrate strong performance in long-sequence modeling and computational efficiency, such as Mamba2 (Gu & Dao, 2023; Dao & Gu, 2024) and GatedDeltaNet (Yang et al., 2024b;a). Our analysis goes beyond final performance to deeply investigate their continual adaptation dynamics. We examine their speed of recovery from dynamic shifts, their resilience against interference from obsolete information, and how different training strategies shape these critical behaviors.

The main contributions of this paper include: 1) We formalize Continual In-Context Reinforcement Learning (CICRL), extending ICRL to address the critical challenge of intra-lifetime non-stationarity; 2) We introduce two novel non-stationary benchmarks for evaluating an agent's ability to adapt to and forget dynamic changes within a single lifetime; 3) We provide a systematic evaluation of modern sequence model architectures on CICRL tasks, revealing their strengths and limitations under non-stationary dynamics; and 4) We offer key insights into the continual learning dynamics of sequence models, highlighting open challenges and future directions for building more robust CICRL agents.

2 Related Work

In-Context Reinforcement Learning. The concept of In-Context Reinforcement Learning (ICRL) is inspired by in-context learning in large language models (Brown et al., 2020; Li et al., 2023; Ruoss et al., 2024). Its core objective is to enable pretrained agents to rapidly adapt to new tasks by leveraging their interaction history, without requiring gradient updates to their network parameters (Laskin et al., 2022; Lee et al., 2023; Lu et al., 2023; Team et al., 2023). Early works such as

RL^2 (Duan et al., 2016) employ Recurrent Neural Networks (RNNs) as memory modules, training agents with standard reinforcement learning to implicitly learn a "fast learning algorithm." More recently, Transformer-based ICRL methods have demonstrated stronger adaptability and generalization potential. For instance, Algorithm Distillation (AD) (Laskin et al., 2022) enables Transformers to imitate the policy improvement process of a source RL algorithm by distilling its learning histories. Decision-Pretrained Transformer (DPT) (Lee et al., 2023) achieves a decision-making mechanism akin to posterior sampling by predicting optimal actions. AMAGO (Grigsby et al., 2023; 2024) successfully applies long-sequence Transformers to end-to-end RL by redesigning the off-policy actor-critic update, showing excellent performance in meta-learning and long-term memory domains. Although these studies represent significant progress in the ICRL field, they primarily focus on how agents adapt to different static instances within a broad task distribution. Less attention has been given to the continual adaptation capabilities of ICRL agents when the environment's dynamics change continuously within an agent's single lifetime (intra-lifetime).

Reinforcement Learning Environments and Non-Stationarity. The design of Reinforcement Learning (RL) environments is crucial for algorithm evaluation and development. Research progress has been driven by a progression from classic environments like Atari (Mnih et al., 2013) and MuJoCo (Todorov et al., 2012), to increasingly complex procedurally generated environments such as Procgen (Cobbe et al., 2020),NetHack (Küttler et al., 2020), XLand (Team et al., 2023), and Kinetix (Matthews et al., 2024b). These settings have continuously pushed research into agent generalization and adaptability. Furthermore, reconstructing environments like XLand-Minigrid (Nikulin et al., 2024) and Craftax (Matthews et al., 2024a) using frameworks such as JAX has significantly enhanced the efficiency of large-scale experiments. However, benchmarks specifically designed for systematically studying ICRL performance under non-stationarity, where the environment itself undergoes continuous unknown changes, remain scarce. Non-stationarity is a prevalent challenge in real-world settings, and while prior works have explored solutions (Hamadanian et al., 2023; Gospodinov et al., 2024), they often involve explicit online adaptation mechanisms. In contrast, our work investigates whether sequence models can address this challenge solely through their inherent context-processing capabilities.

Sequence Models. Sequence models, particularly the Transformer architecture based on self-attention mechanisms (Vaswani et al., 2017), have become powerful tools for processing sequential data. They have achieved revolutionary results in fields such as natural language processing (Brown et al., 2020), computer vision (Dosovitskiy et al., 2020), and reinforcement learning (Chen et al., 2021; Janner et al., 2021). With its parallel processing capabilities and effective capture of long-range dependencies, the Transformer provides a solid model foundation for ICRL. In recent years, a series of novel sequence architectures have been proposed to further enhance the efficiency and performance of long-sequence modeling. For instance, Mamba and its variants (Gu & Dao, 2023; Dao & Gu, 2024) introduce selective state space models, achieving linear-time complexity for long-sequence processing through hardware-friendly designs. DeltaNet and its gated versions (Yang et al., 2024b;a) enhance the model's associative memory and precise update capabilities by incorporating the delta rule. The development of these advanced sequence models offers a diverse range of architectural choices for constructing more powerful CICRL agents.

3 Problem Formulation

In this section, we first present a formal formulation of standard In-Context Reinforcement Learning (ICRL), emphasizing its core assumption of static environments. Building upon this foundation, we introduce and define Continual In-Context Reinforcement Learning (CICRL), which explicitly captures the challenges faced by agents in non-stationary environments.

3.1 In-Context Reinforcement Learning in a Static World

The standard ICRL framework is built upon Partially Observable Markov Decision Processes (POMDPs). Each individual learning task corresponds to a POMDP instance, jointly defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, an observation space $\mathcal{O}$, a transition function T, a reward function R, an observation function Ω, a discount factor γ, and a maximum horizon H.

In ICRL, the agent is exposed to a distribution of tasks $p(\mathcal{M})$. Its objective is to learn a single policy π_θ parameterized by θ. This policy takes a continuous interaction history as context, denoted by $\mathcal{C}_t$, which contains the sequence of all observations, actions, and rewards from the initial time step up to the current time. Based on this context, the policy outputs the next action a_t. When encountering a new task instance sampled from the task distribution, the policy parameters θ remain fixed. The agent relies entirely on its growing context $\mathcal{C}_t$ to infer the characteristics of the current, previously unseen task and to rapidly adapt its behavior. A core assumption of this framework is intra-task stationarity: while an agent must be capable of adapting to different tasks, once a specific task instance $\mathcal{M}_i$ is selected, its underlying environmental dynamics, such as the transition function T_i and reward function R_i, remain fixed throughout the agent's lifetime interacting with it. Consequently, adaptation in ICRL reduces to inferring unknown but constant environmental parameters from the observation history.

3.2 Continual ICRL in a Dynamic World

To investigate an agent's continual adaptation capabilities in a more realistic, ever-changing environments, we introduce the formal framework of Continual In-Context Reinforcement Learning (CICRL). The core feature of CICRL is intra-lifetime non-stationarity: environmental dynamics evolve continuously within a single interaction lifetime of the agent.

Unlike standard ICRL, we no longer assume a fixed distribution of task instances. Instead, the agent engages in a single, extended interaction within a continuously evolving environment. This evolution can be modeled as a sequence of POMDPs, $\{\mathcal{M}^{(k)}\}_{k=1}^{K}$, where during the k-th phase, the environment follows the dynamics of the POMDP instance $\mathcal{M}^{(k)}$, with transition and reward functions $T^{(k)}$ and $R^{(k)}$, respectively.

The fundamental difference from standard ICRL is that the transition from phase k to $k+1$ occurs within the agent's uninterrupted interaction, and the historical context $\mathcal{C}_t$ is never reset. Consequently, the context $\mathcal{C}_t$ may contain information from multiple, potentially conflicting environmental dynamics. As in ICRL, the agent's policy π_θ operates without gradient updates. However, its objective now is to maximize long-term cumulative return across this evolving sequence of POMDPs. This imposes additional demands: the policy must not only infer the current environmental dynamics $\mathcal{M}^{(k)}$ from the context, but also continuously track changes and mitigate interference from outdated information to achieve robust online adaptation.

4 The CICRL Benchmark Suite

To systematically investigate Continual In-Context Reinforcement Learning (CICRL), we construct a benchmark suite comprising two complementary environments, which cover both discrete, rule-based symbolic reasoning domains and continuous, physics-based dynamic control domains. We substantially modified both to exhibit intra-lifetime non-stationarity. This section details the intrinsic mechanics of these environments, our specific methods for introducing non-stationarity, and the algorithmic framework and evaluation protocols used in our experiments.

4.1 Rule-Based Non-Stationarity: The XLand-Minigrid Environment

Our first benchmark builds on XLand-Minigrid (Nikulin et al., 2024), a scalable, JAX-based grid-world environment capable of procedurally generating vast number of logically structured tasks.

Core Mechanics and Agent Interface. In XLand-Minigrid, each task is defined by a set of "Rules" and a "Goal." The rules act as the environment's "physical laws," defining how objects interact, e.g., "placing a blue pyramid ▲ next to a purple square ■ generates a yellow key ⚷." The agent must discover these hidden rules through trial and error and plan a sequence of actions to synthesize the goal object. Observations are symbolic partial grids, with each cell containing object type and color IDs. The action space is discrete, consisting of navigation and interaction commands.

Introducing Intra-Lifetime Non-Stationarity. In standard tasks, the rule-set remains fixed throughout the agent's lifetime. To introduce non-stationarity, we design an intra-lifetime rule evolution protocol, under this protocol, the agent interacts in a single, long session, and every N_x

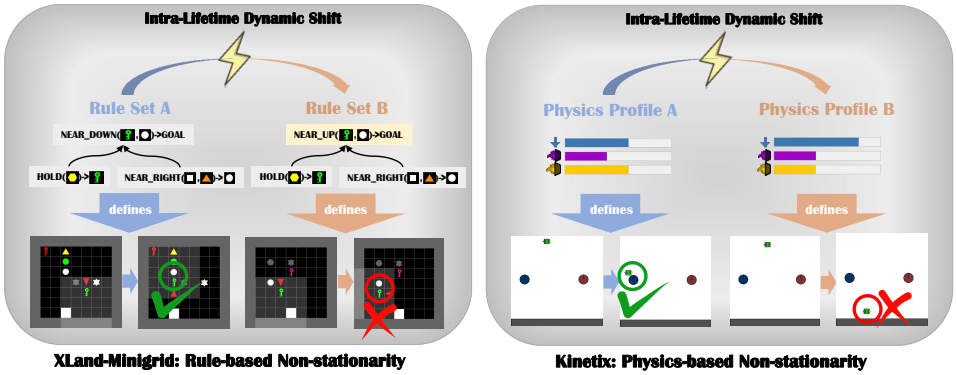

Figure 2: Examples of non-stationary dynamics in the CICRL benchmark suite. Left: In XLand-Minigrid, the environment's symbolic rule set changes intra-lifetime (e.g., the goal condition shifts from NEAR_DOWN to NEAR_UP), causing a previously successful policy to fail. Right: In Kinetix, underlying physical parametersthe underlying physical parameters (e.g., gravity or thruster power) are altered, which also invalidates the agent's learned behavior.

episodes, the environment's rule-set changes. As shown in Figure 2 (left), this evolution includes replacing key rules (e.g., the goal condition changing from 'NEAR_DOWN' to 'NEAR_UP'), as well as adding or removing "Distractor Rules." Agents must therefore continually infer which "physical laws" are currently active while disregarding outdated context.

4.2 Physics-Based Non-Stationarity: The Kinetix Environment

To validate the generalization of CICRL to continuous control domains, our second benchmark builds on Kinetix (Matthews et al., 2024b), a JAX-based 2D physics simulation environment.

Core Mechanics and Agent Interface. A Kinetix scene is procedurally composed of fundamental physical components like rigid bodies, joints, motors and thrusters. The agent controls actuated components (e.g., managing motor torques) to achieve goals such as colliding a green target object collide with a blue goal object. In this environment, agents observe raw visual pixels of the scene and output continuous control signals to drive the motors and thrusters.

Introducing Intra-Lifetime Non-Stationarity. We introduce non-stationarity into the environment on two levels: **structural layout** and **physical dynamics**. First, the layout of objects in the environment changes periodically. Second, we modify the underlying physics engine to enable dynamic changes in the environment's fundamental physical constants during an agent's lifetime. Parameters are randomized via a sampling mechanism at two levels: (i) **global dynamics** that affect the entire scene, such as gravity magnitude, thruster power, motor power, and base friction; and (ii) independent **object-specific properties** for each object, such as density, specific friction coefficients, and coefficients of restitution (elasticity). Every N_k episodes, the system samples a new set of parameters, effectively placing the agent into a novel "physical reality". As illustrated in Figure 2 (right), every N_k episodes, the system samples a new set of parameters (e.g., switching from "Physics Profile A" to "Physics Profile B"), effectively placing the agent into a novel "physical reality."

Training and Test Ruleset. A key feature of Kinetix is the distinction between its training and test sets. The training data is generated entirely through procedural synthesis. The test set, however, is based on a series of human-designed levels that test specific physical reasoning and control skills. To align this test set with our CICRL setting, we apply the same dynamic randomization of physical parameters to these human-designed levels. This evaluates whether agents can transfer dynamics inference abilities learned in procedural training to structured but physically unfamiliar tasks.

4.3 Algorithmic Framework and Evaluation Protocol

To train and evaluate agents on these benchmarks, we employ a unified algorithmic framework and design a specialized evaluation protocol that analyzes their continual adaptation process across multiple dimensions.

Training Framework. Our training pipeline is based on the Proximal Policy Optimization (PPO) algorithm, integrated with a plug-and-play sequence model backbone. Non-stationarity is integrated directly into the training loop: after each dynamic phase, the environment manager samples a new duration $N \sim U(N_{\min}, N_{\max})$ for each parallel environment instance, where $N_{\min}$ and $N_{\max}$ are predefined episode counts. After this duration, the environment automatically switches its dynamic rules or physical parameters. This ensures that agents are continually exposed to dynamically paced changes during training.

Evaluation Protocol. Traditional average return metrics obscure the details of an agent's behavior during dynamic changes. To more finely characterize continual adaptation, we first define some notations. Assume the evaluation process runs in D parallel environments for a total of E episodes. We record all returns in a matrix $\mathbf{R} \in \mathbb{R}^{D \times E}$, where $R_{i,j}$ is the return of the i-th environment in the j-th episode. Let $M(i, j)$ denote the dynamics (e.g., a specific rule set or physics profile) in effect at episode (i, j). Based on this, we design the following suite of evaluation metrics:

• **Zero-Shot Return (ZS-Return):** This metric measures the model's foundational capabilities without any valid historical context. We first compute the average zero-shot return $R_{ZS}(k)$ for each encountered dynamic k, which is the average performance when an agent first encounters dynamic k. The final ZS-Return is the macro-average of all these values. Let $\mathcal{S}$ be the set of all coordinates (i, j) where a dynamic is encountered for the first time: $R_{ZS} = \frac{1}{|\mathcal{S}|} \sum_{(i,j)\in\mathcal{S}} R_{i,j}$

• **Average Return (Avg-Return):** This is the average of all elements in the return matrix $\mathbf{R}$, providing a macroscopic measure of the model's overall performance: $R_{Avg} = \frac{1}{D \cdot E} \sum_{i=1}^{D} \sum_{j=1}^{E} R_{i,j}$.

• **In-Context Delta ($\Delta_{\text{In-Context}}$):** This metric quantifies the average overall impact of contextual information. It is the mean difference between the return $R_{i,j}$ and the zero-shot return of its corresponding dynamic, $R_{ZS}(M(i, j))$: $\Delta_{\text{In-Context}} = \frac{1}{D \cdot E} \sum_{i=1}^{D} \sum_{j=1}^{E} (R_{i,j} - R_{ZS}(M(i, j)))$.

• **Switch Resilience (Δ_{Switch}):** This metric specifically assesses the agent's resilience immediately after a dynamic shift. It measures the average difference between the return in the *first* episode after a switch and the ZS-Return of the new dynamic. Let $\mathcal{S}'$ be the set of all episode coordinates where a dynamic switch occurs (excluding initial episodes): $\Delta_{\text{Switch}} = \frac{1}{|\mathcal{S}'|} \sum_{(i,j)\in\mathcal{S}'} (R_{i,j} - R_{ZS}(M(i, j)))$. A non-negative Δ_{Switch} value indicates that the agent can effectively ignore outdated information and is robust to abrupt shifts.

• **Adaptation Gain (Δ_{Adapt}):** This metric measures the agent's learning ability within a stable dynamic. It is the average gain in return for all *non-switch* episodes (i.e., the second episode and beyond within a dynamic), relative to the ZS-Return of that dynamic. Let $\mathcal{A}$ be the set of all non-switch episode coordinates: $\Delta_{\text{Adapt}} = \frac{1}{|\mathcal{A}|} \sum_{(i,j)\in\mathcal{A}} (R_{i,j} - R_{ZS}(M(i, j)))$.

Through this decoupled evaluation protocol, we can analyze the model's behavior from different dimensions, not only assessing its final performance but also revealing its learning, forgetting, and interference-resistance mechanisms when facing continual dynamic changes.

5 Experiments

5.1 Experimental Setup

Environments and Training Protocols. Our experiments are conducted in two distinct non-stationary environments: the rule-based **XLand-Minigrid** and the physics-based **Kinetix**. In XLand-Minigrid, the agent engages in a total of 10 billion (10B) interaction steps, while in the more computationally demanding Kinetix environment, the interaction budget is set to 1 billion (1B) steps. During both training and testing under non-stationary conditions, the frequency of environmental dynamic changes is set to occur after a random number of episodes, sampled uniformly from 1 to 10. This stochastic change frequency is designed to mimic real-world scenarios where the pace of environmental evolution is irregular. Unless otherwise stated in ablation studies, the interaction context length for all models during training is fixed at 25600 steps.

Models. We evaluate a diverse set of advanced sequence models, including Transformer, Mamba2, and GatedDeltaNet. To ensure a fair comparison, we configure each model with a similar number of layers and width. We employ a unified PPO training framework without introducing any architecture-specific modifications or enhancements. This allows us to directly assess the models' intrinsic continual adaptation abilities in the CICRL setting.

Table 1: Performance on **XLand-Minigrid**. Models trained under Static and Non-Static paradigms are evaluated in both static and non-static test environments.

Model	ZS-Return	Static Env. Eval.		Non-Static Env. Eval.			
		Avg-Return	$\Delta_{\text{In-Context}}$	Avg-Return	$\Delta_{\text{In-Context}}$	Δ_{Switch}	Δ_{Adapt}
GatedDeltaNet (Static)	0.084	0.188	**0.100**	0.081	-0.003	-0.012	0.000
GatedDeltaNet (Non-Static)	0.173	**0.221**	0.044	0.211	**0.038**	0.025	0.044
Mamba2 (Static)	0.089	0.166	0.078	0.093	0.007	-0.012	0.009
Mamba2 (Non-Static)	**0.190**	0.216	0.027	**0.213**	0.027	0.009	0.028
Transformer (Static)	0.030	0.054	-0.002	0.050	-0.002	0.021	0.020
Transformer (Non-Static)	0.109	0.179	0.000	0.174	0.001	**0.058**	**0.067**

Table 2: Performance on **Kinetix**. The evaluation setup mirrors that of Table 1.

Model	ZS-Return	Static Env. Eval.		Non-Static Env. Eval.			
		Avg-Return	$\Delta_{\text{In-Context}}$	Avg-Return	$\Delta_{\text{In-Context}}$	Δ_{Switch}	Δ_{Adapt}
GatedDeltaNet (Static)	0.023	-0.020	-0.019	-0.023	-0.002	-0.038	-0.051
GatedDeltaNet (Non-Static)	0.034	**0.015**	0.002	-0.004	0.021	-0.041	-0.038
Mamba2 (Static)	**0.035**	0.005	-0.005	-0.003	**0.024**	-0.042	-0.038
Mamba2 (Non-Static)	-0.065	-0.080	**0.004**	-0.008	-0.070	**0.057**	**0.056**
Transformer (Static)	-0.050	-0.059	-0.002	**0.000**	-0.057	0.052	0.051
Transformer (Non-Static)	0.044	**0.015**	0.000	-0.004	**0.034**	-0.052	-0.048

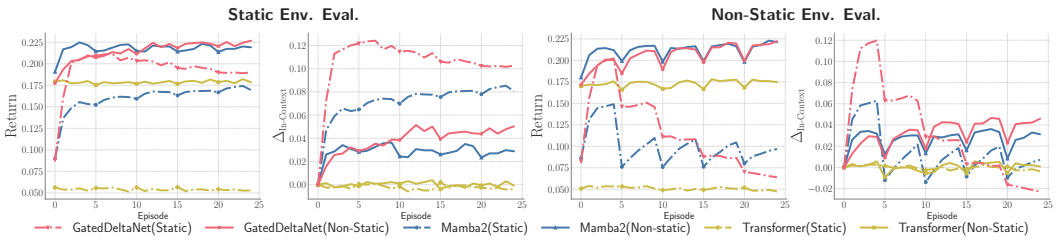

Figure 3: Results on XLand-Minigrid. In the non-stationary evaluation (right), models trained with non-stationarity (solid lines) successfully adapt to dynamic changes, while statically trained models (dashed lines) exhibit sustained low performance.

5.2 Main Results: Static vs. Non-Stationary Training

To systematically evaluate the necessity of non-stationary training, we compare models trained under **Static** and **Non-Stationary** paradigms. All models are evaluated in both static and non-stationary test environments, with detailed results presented in Table 1 and Table 2.

In the **XLand-Minigrid** environment, the advantages of non-stationary training are clearly validated, with the performance of GatedDeltaNet being particularly outstanding. As shown in Table 1, after non-stationary training, GatedDeltaNet's average return (Avg-Return) in the non-stationary test surged from 0.081 to 0.211, a significant increase that places it at a top-tier level among all models. The dynamic curves in Figure 3 visually illustrate this point: in the non-stationary evaluation (right plots), statically trained models (dashed lines) suffer a performance collapse after an environmental shift, whereas the non-stationarily trained GatedDeltaNet (solid red line) demonstrates strong recovery capabilities and consistently maintains a high level of performance. Although Mamba2 achieves a slightly higher average return, GatedDeltaNet shows highly competitive performance, proving that its architecture is well-suited for handling dynamic changes in rule-based, symbolic reasoning tasks.

In the more challenging physics-based **Kinetix** environment, all models faced greater online adaptation pressure, yet GatedDeltaNet once again proved its robustness. As shown in Table 2, the difficulty of this task resulted in negative adaptation gains (Δ_{Adapt}) for most models, indicating that context sometimes acted as a source of interference. However, under these demanding conditions, GatedDeltaNet achieved the highest average return (Avg-Return) alongside Transformer, showcasing its overall policy robustness in dealing with complex physical dynamics. Furthermore, it obtained a robust in-context gain ($\Delta_{\text{In-Context}}$), indicating its ability to effectively leverage contextual

information. While Mamba2 exhibited unique performance on immediate post-switch adaptation metrics, GatedDeltaNet's ability to ensure strong, comprehensive long-term performance makes it an extremely reliable choice.

5.3 Ablation Studies: Key Factors Influencing Continual Adaptation

Building on the results above, we conduct a series of ablation studies in the non-stationary XLand-Minigrid environment, primarily using the GatedDeltaNet architecture. These experiments aim to identify the key hyperparameters and training factors that affect continual adaptation.

Impact of Environmental Change Frequency. We first examine how the frequency of environmental changes during training affects generalization to unseen non-stationary patterns. We train models with various frequencies: fast-random (1-5 episodes), slow-random (10-20 episodes), fixed (5 or 10 episodes), and our default balanced-random frequency (1-10 episodes). The results are shown in Figure 4. We evaluate all models in both a fast-changing (test frequency of 1) and a slow-changing (test frequency of 10) environment. The results highlight the importance of randomness. Models trained with a fixed frequency overfit to a specific rhythm and fail to generalize well to different change speeds. Among the randomized settings, overly rapid changes (1-5 episodes) may not allow the agent to fully learn from each dynamic, while overly slow changes (10-20 episodes) reduce the diversity of adaptation experiences. A randomized frequency of 1-10 episodes achieves the best trade-off, yielding the most robust performance across both fast and slow test scenarios.

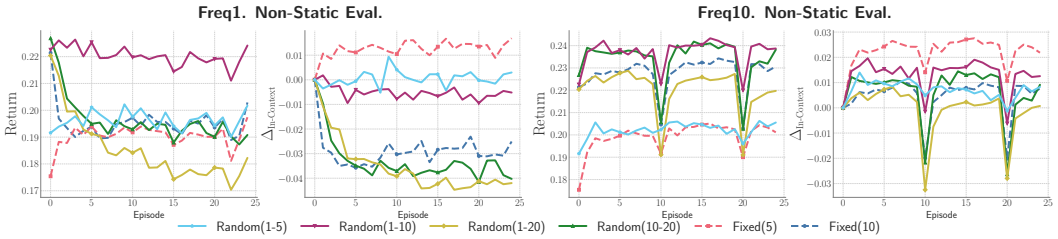

Figure 4: Impact of training with different dynamic change frequencies. Models are trained with fixed or randomized change frequencies and evaluated in environments with fast (Freq1, left) and slow (Freq10, right) changes. Training with a randomized frequency (Random 1-10) yields the most robust performance across both test settings.

The Role of Context Length. Context length is central to ICRL. To understand its impact in the CICRL setting, we train separate models using different context lengths: 6400, 12800, 25600, and 51200 steps. The results, shown in Figure 5, reveal that longer is not always better. Performance peaks at a context length of 25600 before degrading at 51200. From a training dynamics perspective, extremely long sequences can exacerbate the credit assignment problem for the PPO algorithm. The gradients from recent, more relevant transitions may be diluted, which could destabilize the learning process or necessitate adjustments to hyperparameters like the learning rate. The plots, which compare performance early (Episodes 1-50) and late (Episodes 151-200), confirm this trend is stable over long horizons. This finding suggests a "sweet spot" for context length and provides important practical guidance for designing CICRL agents.

Scaling Effects of Training. We further investigate how scaling along different dimensions influences model performance. Figure 6 presents the results for scaling interaction data and model size. **Interaction Data:** The left plots show a clear positive correlation between the amount of training data and performance. As the training budget increases from 1B to 20B steps, the model's average return and its ability to adapt after a switch steadily improve. **Model Size:** Similarly, the right plots show that increasing model depth from 3 to 6 layers leads to a significant and consistent improvement in continual adaptation. These results reveal a clear scaling trend, underscoring that CICRL benefits from scaling both data and model size.

Composite Change Patterns: "Stable" vs. "Chaotic" Eras. In the Static-to-Fast setting (Figure 7, left), the environment remains stable for the first 25 episodes before entering a phase of rapid changes. The results show that while performance drops at the moment of the switch, the models quickly adapt to the new chaotic environment without collapsing. In the Fast-to-Static setting

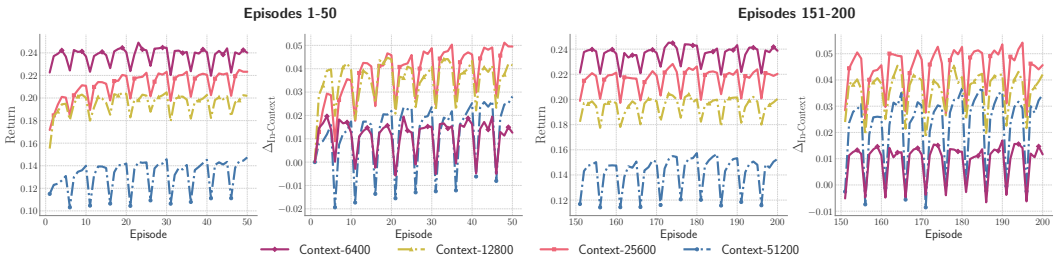

Figure 5: Effect of context length on continual adaptation. The plots show performance at the beginning (Episodes 1-50) and later (Episodes 151-200) of a long evaluation run.

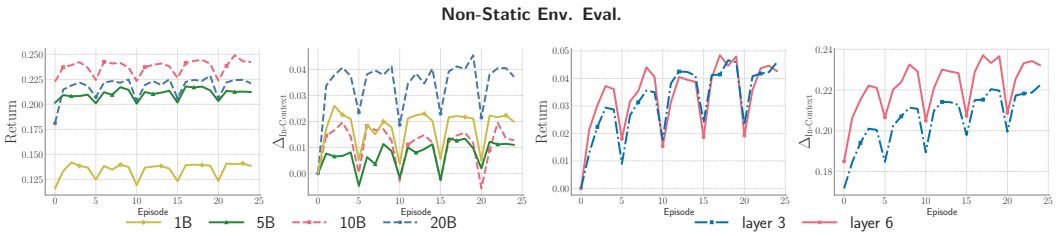

Figure 6: Scaling effects of training data and model size. **Left:** Performance improves as the total number of training interaction steps increases from 1B to 20B. **Right:** A deeper model (6 layers) consistently outperforms a shallower one (3 layers).

(Figure 7, right), the environment transitions from frequent changes to long-term stability. We observe that after the environment stabilizes at episode 25, the models' performance and in-context gains significantly improve and consolidate, demonstrating their ability to effectively exploit the new regularity.

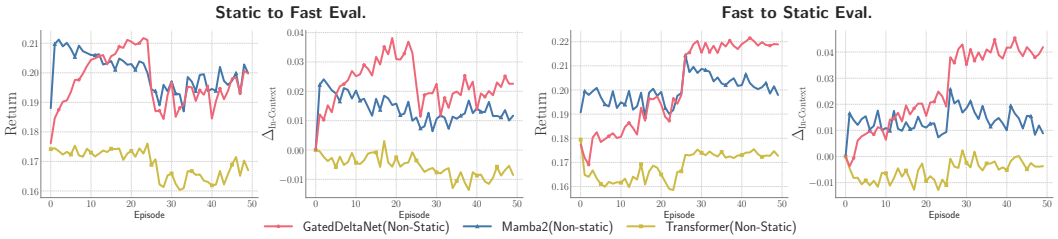

Figure 7: Robustness to composite dynamic changes. **Left (Static to Fast):** The environment is stable for the first 25 episodes before entering a phase of rapid changes. **Right (Fast to Static):** The environment starts with rapid changes and becomes stable after 25 episodes.

6 Conclusion

In this paper, we formally introduced and systematically investigated Continual In-Context Reinforcement Learning (CICRL), extending the ICRL paradigm from static-task adaptation to the more realistic and challenging setting of intra-lifetime non-stationarity. Using our purpose-built non-stationary benchmark suite, we show that non-stationary training is crucial for overcoming contextual catastrophic forgetting and achieving robust continual adaptation. Our experiment further highlight that linear attention architectures, such as GatedDeltaNet, consistently outperform standard Transformers under dynamic conditions. Finally, our ablation studies reveal the critial role of factors such as context length and environmental change frequency, offering concrete insights for designing CICRL agents capable of adapting in complex, evolving environments.

Ethics Statement

This work focuses exclusively on synthetic and simulated environments—specifically symbolic reasoning and physics-based control domains—and does not involve human subjects, personal data, or sensitive content. Our research aims to advance the fundamental understanding of continual in-context reinforcement learning, without proposing direct real-world deployments that may carry immediate societal risks. Nevertheless, we acknowledge that reinforcement learning systems capable of continual adaptation could be applied in safety-critical domains. To mitigate potential risks, we emphasize that our contributions are intended as controlled benchmarks and analyses for academic research, not as ready-to-deploy systems. We are committed to releasing all code and benchmarks in an open and transparent manner to support reproducibility and responsible use. No conflicts of interest, external sponsorship, or ethical concerns beyond those discussed above are associated with this work.

Reproducibility Statement

We have taken several steps to ensure the reproducibility of our work: 1) For the benchmark suite, we provide detailed descriptions of the two non-stationary benchmark domains in the main paper and the supplementary material, including their environment dynamics, modification procedures, and change schedules. 2) For models and training, we specify the model architectures, parameterization strategies, training protocols, and hyperparameters in the paper. 3) We will release code and the benchmark environments to ensure reproducibility of our experiments. 4) All reported metrics and evaluation protocols are explicitly defined. Through these measures, we aim to make our work fully reproducible and to facilitate further research on continual in-context reinforcement learning.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl˜2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Emiliyan Gospodinov, Vaisakh Shaj, Philipp Becker, Stefan Geyer, and Gerhard Neumann. Adaptive world models: Learning behaviors by latent imagination under non-stationarity. *arXiv preprint arXiv:2411.01342*, 2024.

Jake Grigsby, Linxi Fan, and Yuke Zhu. Amago: Scalable in-context reinforcement learning for adaptive agents. *arXiv preprint arXiv:2310.09971*, 2023.

Jake Grigsby, Justin Sasek, Samyak Parajuli, Daniel Adebi, Amy Zhang, and Yuke Zhu. Amago-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers. *Advances in Neural Information Processing Systems*, 37:87473–87508, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Pouya Hamadanian, Arash Nasr-Esfahany, Malte Schwarzkopf, Siddartha Sen, and Mohammad Alizadeh. Online reinforcement learning in non-stationary context-driven environments. *arXiv preprint arXiv:2302.02182*, 2023.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. *Advances in Neural Information Processing Systems*, 33:7671–7684, 2020.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:43057–43083, 2023.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, pp. 19565–19594. PMLR, 2023.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*, 2023.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:47016–47031, 2023.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Jackson, Samuel Coward, and Jakob Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. *arXiv preprint arXiv:2402.16801*, 2024a.

Michael Matthews, Michael Beukman, Chris Lu, and Jakob Foerster. Kinetix: Investigating the training of general agents through open-ended physics-based control tasks. *arXiv preprint arXiv:2410.23208*, 2024b.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning. *arXiv preprint arXiv:2502.07978*, 2025.

Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. Xland-minigrid: Scalable meta-reinforcement learning environments in jax. *Advances in Neural Information Processing Systems*, 37:43809–43835, 2024.

Anian Ruoss, Fabio Pardo, Harris Chan, Bonnie Li, Volodymyr Mnih, and Tim Genewein. Lmact: A benchmark for in-context imitation learning with long multimodal demonstrations. *arXiv preprint arXiv:2412.01441*, 2024.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, et al. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*, 2023.

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*, pp. 5026–5033. IEEE, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv:2406.06484*, 2024b.

A USE OF LARGE LANGUAGE MODELS

In this work, the large language model (LLM) is employed exclusively for text polishing purposes. Its role is limited to refining the linguistic quality, coherence, and stylistic consistency of the textual content, without involvement in data generation, analytical reasoning, or substantive content creation.