NAMI: A HEURISTIC FORWARD TRANSFER APPROACH FOR CONTINUAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual reinforcement learning (CRL) requires a reinforcement learning agent to adapt rapidly to new tasks by reusing knowledge acquired from previous ones, a capability known as forward transfer. Existing architecture-based methods, which typically maintain task-specific modules and reuse them for future tasks, often perform forward transfer at the level of entire networks. While effective, such coarse-grained transfer entangles reusable and task-specific modules, making it difficult to reuse prior knowledge flexibly under strict interaction budgets. In this paper, we propose Neuron-wise mAnhattan-norm Maximization Initialization (NAMI), a simple and efficient strategy for forward transfer in CRL. Instead of transferring the entire policy networks, NAMI initializes each neuron of a new task policy network by selecting the corresponding neuron with the largest Manhattan norm (i.e., the maximum sum of absolute weights) from the previously learned policy networks. This converts model-level transfer into neuron-level selection, allowing the new policy to inherit salient reusable components from different prior tasks while remaining a single compact network whose inference cost is independent of the number of previous tasks. Extensive experiments show that NAMI achieves superior performance over various state-of-the-art methods.

1 INTRODUCTION

Continual reinforcement learning (CRL) (Khetarpal et al., 2022) studies the problem of training a reinforcement learning agent (Xie et al., 2024; Ghosh et al., 2023; Li et al., 2023) on a sequence of tasks, in which the agent is expected to adapt to new tasks while retaining and reusing knowledge acquired from previous ones. While CRL has traditionally been associated with mitigating forgetting on previously learned tasks, an equally important objective is to promote forward transfer, where prior knowledge is reused to accelerate learning on future tasks. Forward transfer is especially critical when environment interaction is expensive, and each new task must be learned under a limited training budget. Due to its practical importance, many CRL methods (Rusu et al., 2016; Rolnick et al., 2019; Kirkpatrick et al., 2017) have been developed to improve continual adaptation across tasks.

Among these methods, architecture-based approaches (Ben-Iwhiwhu et al., 2023; Mallya & Lazebnik, 2018; Rusu et al., 2016) are particularly intuitive and interpretable. A common strategy is to allocate, preserve, or compose task-specific modules so that knowledge learned from previous tasks can be retained and reused when solving new ones. By maintaining separate modules for different tasks, these approaches naturally mitigate forgetting. More importantly, the preserved modules also serve as reusable knowledge sources for forward transfer to future tasks. In this sense, the primary objective of these approaches is not to prevent forgetting but to reuse previously learned modules to accelerate learning on future tasks. Therefore, recent methods (Hu et al., 2025; Malagón et al., 2024) have increasingly focused on how to achieve efficient forward transfer under limited interaction budgets.

Despite this progress, the efficiency of existing forward-transfer methods remains limited. As illustrated in Figure 1(a), existing methods typically perform forward transfer at the level of entire policy networks, either in the output space or in the parameter space. Output-space methods (Rusu et al., 2016; Malagón et al., 2024) reuse previous policy outputs, but their reliance on all historical networks leads to redundant computation. To reduce this overhead, parameter-space methods (Hu et al., 2025; Dohare et al., 2024) merge previous policy networks, but conflicting parameter directions across tasks can cause the cancellation of useful components. Overall, such coarse-grained transfer

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

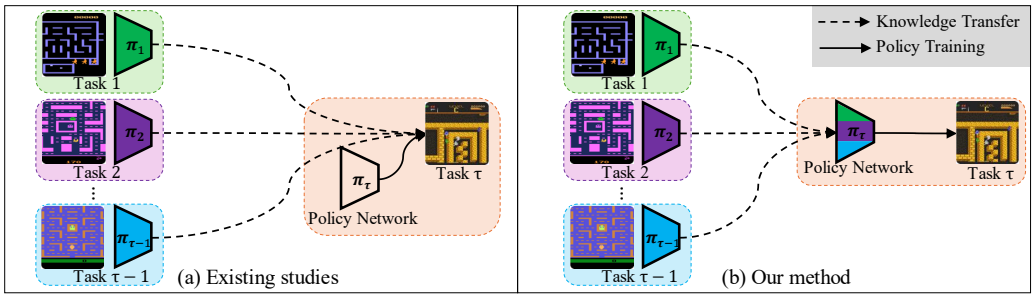


Figure 1: Illustration of our work. (a) Existing approaches leverage the whole policy network of old tasks to transfer knowledge for the current task. (b) Different from existing studies, our proposed NAMI selects neurons from previous networks based on the parameter score for forward transfer.

can be inefficient due to fine-grained knowledge redundancy and knowledge conflict among previous policy networks. Therefore, efficient forward transfer in CRL requires a more fine-grained and flexible mechanism without relying on whole-network transfer.

To this end, we propose Neuron-wise mAnhattan-norm Maximization Initialization (NAMI), a novel forward knowledge transfer method for CRL. As illustrated in Figure 1(b), for a new task τ, NAMI initializes the policy network π_τ by selecting neurons from previous task-specific networks through a binary mask. Specifically, we first formulate fine-grained forward transfer as a combinatorial optimization problem in parameter space. Through empirical analysis, we observe that neurons with larger Manhattan norms tend to encode more reusable task-specific modules, where the Manhattan norm is defined as the sum of absolute weights. This observation motivates a simple surrogate: for each neuron in the new policy, NAMI selects the corresponding neuron with the maximum Manhattan norm across all past policies. By selecting strong neurons instead of mixing whole models, NAMI alleviates destructive interference among tasks and preserves informative reusable representations. The resulting policy remains a single compact network, whose inference cost is independent of the number of previous tasks. Overall, NAMI consolidates knowledge from multiple past tasks and provides a strong initialization for the new task, thereby accelerating forward transfer in CRL.

Our main contributions can be summarized as:

1) We identify a key limitation of existing architecture-based CRL methods, namely coarse-grained transfer, and highlight the need for a more fine-grained and flexible knowledge reuse mechanism.

2) We propose NAMI, a neuron-level forward transfer framework that initializes each neuron of a new task using the corresponding neuron with the largest Manhattan norm across previous tasks.

3) We conduct extensive experiments, and the results show that NAMI improves forward transfer under limited interaction budgets over representative architecture-based baselines.

2 Related Work

Architecture-based approaches. Architecture-based methods mitigate interference by adapting the network structure across tasks. ProgNet (Rusu et al., 2016) transfers prior features via lateral connections, while PackNet (Mallya & Lazebnik, 2018) packs multiple tasks into a single network through iterative pruning and retraining. MaskNet (Ben-Iwhiwhu et al., 2023) and CReLUs (Abbas et al., 2023) improve parameter reuse and plasticity by learning task-specific masks or modifying activation functions. CbpNet (Dohare et al., 2024) periodically reinitializes a subset of units, while ReGraMa (Liu et al., 2026) uses gradient information to identify and reset inactive neurons. CompoNet (Malagón et al., 2024) selectively combines previous policies with its internal policy, accelerating learning on a new task. CKA-RL (Hu et al., 2025) maintains a task-specific knowledge vector pool and dynamically uses the vectors to adapt the agent to new tasks.

Other approaches. Other approaches mainly include regularization-based and rehearsal-based methods. Regularization-based approaches (Muppidi et al., 2024; Kirkpatrick et al., 2017; Kessler et al., 2020) constrain parameter updates to reduce interference between old and new knowledge while

maintaining learning plasticity. For instance, ITER (Igl et al., 2021) employs knowledge distillation to constrain parameter changes. Fast TRAC (Muppidi et al., 2024) introduces a parameter-free optimization mechanism to regulate gradient updates under distribution shifts, while C-CHAIN (Tang et al., 2025) reduces network churn and adaptively controls gradient updates to mitigate plasticity loss. Rehearsal-based approaches (Rolnick et al., 2019; Lin et al., 2023b;a) preserve past knowledge by replaying stored or regenerated experiences. DRAGO (Fu et al., 2025) employs generative replay to recover past experiences, while FAME (Sun et al., 2025) integrates previous experience through coupled fast and meta learners for efficient knowledge adaptation.

Our approach. Our method is an architecture-based approach. Different from existing model-level transfer methods, our method performs transfer at the neuron-level. By recombining individual neurons across previously learned policies, our approach enables a more fine-grained and flexible form of knowledge reuse for CRL, allowing more effective exploitation of transferable knowledge.

3 PRELIMINARIES

3.1 CONTINUAL REINFORCEMENT LEARNING

Reinforcement learning (Sutton et al., 1998) learns to make sequential decisions and defines this goal as a Markov Decision Process (MDP). The MDP $M = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ is defined by the space of states $\mathcal{S}$, the space of actions $\mathcal{A}$, the state transition probability function $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and the discount factor $\gamma \in [0, 1]$. With the policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, the agent receives the current state $s_t \in \mathcal{S}$ from the environment, and takes an action $a_t \in \mathcal{A}$ sampled from $\pi(\cdot|s_t)$ at each timestep $t \in \mathbb{N}$. Then the environment transitions to a new state $s_{t+1} \sim p(\cdot|s_t, a_t)$ and the reward $r(s_t, a_t)$ can be obtained. During this process, the objective of RL is to find an optimal policy π^* that maximizes the expected sum of discounted rewards for all states as

$$\max_{\pi} \mathbb{E}_{\pi,p}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]. \tag{1}$$

The problem can be extended to CRL if agents are required to learn a sequence of tasks continuously. Given a sequence of N tasks, each task $\tau \in \{1, 2, \ldots, N\}$ is denoted as a stationary MDP

$$M^{(\tau)} = \langle \mathcal{S}^{(\tau)}, \mathcal{A}^{(\tau)}, p^{(\tau)}, r^{(\tau)}, \gamma^{(\tau)} \rangle, \tau = 1, 2, ..., N. \tag{2}$$

Following prior works (Abel et al., 2023; Malagón et al., 2024; Hu et al., 2025), the agent is given only a limited budget of training steps $\Delta^{(\tau)}$ to interact with the task $M^{(\tau)}$ during the learning process. Under this constraint, learning each new task from scratch may lead to slow and inefficient adaptation. Therefore, a key objective of CRL is to accelerate and improve the learning of the current policy $\pi^{(\tau)}$ by effectively leveraging knowledge acquired from previously learned policies $\{\pi^{(i)}\}_{i=1}^{\tau-1}$.

3.2 COARSE-GRAINED TRANSFER IN PARAMETER SPACE

Some coarse-grained methods can realize forward transfer in parameter space. For convenience, we parameterize the policy as a neural network and let $\theta^\tau \in \mathbb{R}^d$ denote the parameters of policy network for task τ. **Fine-tuning** (Wołczyk et al., 2021) initializes from $\theta^{(\tau-1)}$ and maximizes the expected return $L^{(\tau)}(\theta)$ by performing gradient-based updates on task τ as $\theta^{(\tau)} = \theta^{(\tau-1)} + \gamma \nabla_\theta L^{(\tau)}(\theta)$. Here, we still retain $\theta^{(\tau-1)}$ for solving task $\tau - 1$.

Let θ_{base} be a shared base parameterization and $\{v_i\}_{i<\tau}$ be a pool of knowledge vectors acquired from previously learned tasks. For task τ, **CKA-RL** (Hu et al., 2025) constructs $\theta^{(\tau)} = \theta_{\text{base}} + \sum_{i<\tau} \alpha_i^{(\tau)} v_i + v_\tau$, where $\alpha^{(\tau)} = \text{softmax}(\beta^{(\tau)})$ and $\beta^{(\tau)}$ are learnable hyper-parameters.

4 METHODOLOGY

In this section, we present our method for CRL. The goal is to construct a strong network initialization for the current task by selectively reusing parameters from previous networks $\{\theta_i\}_{i<\tau}$.

4.1 Fine-grained Transfer in Parameter Space

To construct an initialization for task τ, we introduce parameter-level mixing coefficients $\alpha_{i,j}^{(\tau)}$, generated from learnable hyper-parameters $\beta_{i,j}^{(\tau)}$ via a softmax as

$$\alpha_{i,j}^{(\tau)} = \frac{\exp\left(\beta_{i,j}^{(\tau)}/T\right)}{\sum_{k<\tau}\exp\left(\beta_{k,j}^{(\tau)}/T\right)}, \quad \forall j \in [d],\ \forall i < \tau, \tag{3}$$

where $T > 0$ is a temperature. Then, the parameter-level mixing coefficients can be denoted as

$$\theta_{0,j}^{(\tau)}(\alpha) = \sum_{i<\tau}\alpha_{i,j}^{(\tau)}\,\theta_{i,j}, \quad \sum_{i<\tau}\alpha_{i,j}^{(\tau)} = 1, \quad \alpha_{i,j}^{(\tau)} \geq 0, \quad \forall j \in [d], \tag{4}$$

where $\theta_0^{(\tau)}$ is the initialization for task τ and $\alpha_{i,j}^{(\tau)}$ assigns, for each coordinate j, how much we reuse from historical policy network i. However, directly optimizing $\alpha_{i,j}^{(\tau)}$ is impractical under strict interaction budgets due to the $O((\tau-1)d)$ additional variables. Moreover, soft mixing may dilute salient directions and cause destructive interference between conflicting task-specific parameters.

To avoid soft mixing, we encourage sparse coefficients so that each coordinate is dominated by a single expert. When $T \to 0$ and the maximum logit is unique, $\alpha_{\cdot,j}^{(\tau)}$ becomes one-hot as

$$\lim_{T\to 0}\alpha_{i,j}^{(\tau)} = \mathbf{1}\left[i = \arg\max_{k<\tau}\beta_{k,j}^{(\tau)}\right]. \tag{5}$$

In this way, our target is changed into a parameter-wise combinatorial optimization problem as

$$\begin{aligned} &\max_{\alpha} \quad L^{(\tau)}\left(\theta_0^{(\tau)}(\alpha)\right) \\ \text{s.t.} \quad &\theta_{0,j}^{(\tau)}(\alpha) = \sum_{i<\tau}\alpha_{i,j}^{(\tau)}\theta_{i,j}, \quad \alpha_{i,j}^{(\tau)} \in \{0,1\}, \quad \sum_{i<\tau}\alpha_{i,j}^{(\tau)} = 1, \quad \forall j \in [d]. \end{aligned} \tag{6}$$

Since $\alpha_{\cdot,j}^{(\tau)}$ is a one-hot vector, it induces a unique selected source for each coordinate

$$\theta_{0,j}^{(\tau)} = \theta_{i^{(j)},j}, \quad i^{(j)} = \arg\max_{i<\tau}\alpha_{i,j}^{(\tau)}. \tag{7}$$

4.2 Surrogate Solution

Optimizing the combinatorial problem in Eq. (6) remains difficult under strict interaction budgets, even with fixed step sizes. Therefore, we seek a deterministic and computationally efficient surrogate rule to instantiate the winner $i^{(j)}$.

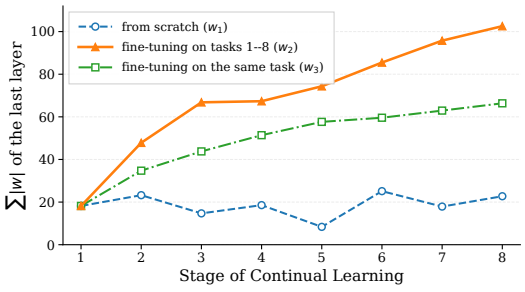


Figure 2: The sum of absolute values of the last-layer parameters with different settings.

To this end, we conduct a heuristic empirical analysis on an eight-task Atari (Machado et al., 2018) sequence. As shown in Figure 2, independently training each task from scratch ($\mathbf{w}_1$) yields relatively small and fluctuating parameter magnitudes, whereas sequential fine-tuning across Tasks 1–8 ($\mathbf{w}_2$) leads to substantial parameter accumulation. To rule out the effect of additional optimization steps, we repeatedly fine-tune on the same task for the same number of stages ($\mathbf{w}_3$). Although $\mathbf{w}_3$ also grows gradually, its magnitude remains considerably smaller than that of $\mathbf{w}_2$, suggesting that heterogeneous task adaptation induces additional parameter accumulation beyond repeated optimization. This observation motivates us to reconsider whole-network transfer and investigate whether transferable knowledge is concentrated in a subset of high-magnitude parameters.

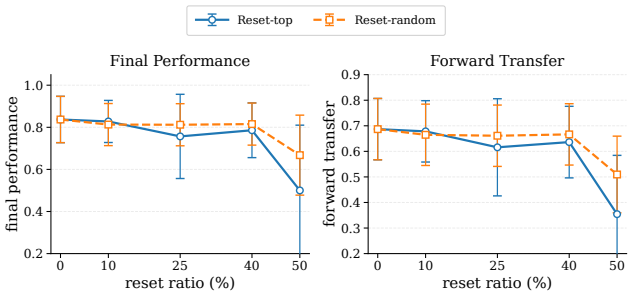


Figure 3: The model performance under different settings.

We further investigate whether parameters with larger magnitudes are more important for forward transfer. Specifically, we rank parameters by their absolute values and compare resetting the parameters with the largest magnitudes (Reset-top) against resetting an equal proportion of randomly selected parameters (Reset-random). As shown in Figure 3, Reset-top generally causes a larger degradation in both final performance and forward transfer, with the gap becoming more evident as the reset ratio increases. This suggests that transferable knowledge is disproportionately concentrated in high-magnitude parameters.

These results do not imply that parameter magnitude is a universally sufficient measure of transferability. Instead, under a strict interaction budget, it provides a simple and low-cost proxy for parameter saliency. We therefore approximate the intractable objective in Eq. (6) using a greedy parameter-wise selection rule: for each coordinate j, we select the previous task whose corresponding parameter has the largest absolute magnitude and use that parameter to initialize the new policy as

$$\mathrm{Score}_{i,j} \triangleq |\theta_{i,j}|, \quad i^{(j)} = \arg\max_{i<\tau} \mathrm{Score}_{i,j}, \quad \theta_{0,j}^{(\tau)} = \theta_{i^{(j)},j}. \tag{8}$$

We refer to this procedure as the *parameter-wise absolute maximization* rule. However, Eq. (8) treats each coordinate independently, whereas parameters within the same neuron are correlated and jointly implement a coherent feature (Wan & Yang, 2025). To preserve such structural dependencies and avoid fragmented initialization, we extend the selection from the parameter level to the neuron level.

Let $\mathcal{G}$ denote a partition of the policy network parameters into neuron groups, and let $\theta_{i,g}$ denote the parameter block associated with group g in historical policy θ_i. Following the same greedy principle, we perform selection at the group level. Specifically, for each $g \in \mathcal{G}$, we measure its saliency using the Manhattan norm (Alzorba et al., 2017), select the historical policy with the largest group-level magnitude, and transfer the entire parameter block to initialize the new policy as

$$\mathrm{Score}_{i,g} \triangleq |\theta_{i,g}|_1 = \sum_{j\in g} |\theta_{i,j}|, \quad i^{(g)} = \arg\max_{i<\tau} \mathrm{Score}_{i,g}, \quad \theta_{0,g}^{(\tau)} = \theta_{i^{(g)},g}, \quad \forall g \in \mathcal{G}. \tag{9}$$

We refer to this procedure as the *neuron-wise absolute maximization* rule. Compared with parameter-wise selection, it transfers coherent parameter blocks rather than independently selected coordinates, thereby preserving neuron structure and reducing the fragmentation of the constructed initialization.

4.3 A Local Optimization Perspective on Transferability

We further examine neuron transferability from a local optimization perspective by considering how transferring a historical neuron affects the objective of the current task. Specifically, we consider a local approximation of the current task objective around a reference parameter $\bar{\theta}$ as

$$L^{(\tau)}(\bar{\theta}+\delta) \approx L^{(\tau)}(\bar{\theta}) + \sum_{j=1}^{d} g_j^{(\tau)}\delta_j - \frac{1}{2}\sum_{j=1}^{d} h_j^{(\tau)}\delta_j^2, \tag{10}$$

where $g^{(\tau)} = \nabla_\theta L^{(\tau)}(\bar{\theta})$ denotes the gradient of the current task and $h_j^{(\tau)} \geq 0$ denotes a diagonal curvature proxy. For neuron group g, let $\delta_{i,g} = \theta_{i,g} - \bar{\theta}_g$ denote the parameter displacement.

A natural first-order transfer criterion that favors historical neurons whose parameter displacement is locally aligned with the gradient can be denoted as

$$S_{i,g}^{\mathrm{FO}} = \left(g_g^{(\tau)}\right)^\top \delta_{i,g} = \sum_{j\in g} g_j^{(\tau)}\delta_{i,j}, \qquad i_{\mathrm{FO}}^{(g)} = \arg\max_{i<\tau} S_{i,g}^{\mathrm{FO}}. \tag{11}$$

Meanwhile, a second-order transfer criterion that additionally penalizes parameter displacements along directions with high local curvature can be denoted as

$$S_{i,g}^{\mathrm{SO}} = \left(g_g^{(\tau)}\right)^\top \delta_{i,g} - \frac{1}{2}\sum_{j\in g} h_j^{(\tau)}\delta_{i,j}^2, \qquad i_{\mathrm{SO}}^{(g)} = \arg\max_{i<\tau} S_{i,g}^{\mathrm{SO}}. \tag{12}$$

Algorithm 1 Neuron-wise mAnhattan-norm Maximization Initialization (NAMI)

Require: Tasks $\{\tau\}_{\tau=1}^{N}$; interaction budgets $\{\Delta^{(\tau)}\}_{\tau=1}^{N}$; neuron groups $\mathcal{G}$; reinforcement learner PPO.
Ensure: Policies $\{\theta^{(\tau)}\}_{\tau=1}^{N}$.
1: Randomly initialize $\theta_0^{(1)}$
2: $\theta^{(1)} \leftarrow \mathrm{PPO}\left(\theta_0^{(1)}, \Delta^{(1)}\right)$
3: $\theta_0 \leftarrow \theta^{(1)}$
4: **for** $\tau = 2$ **to** N **do**
5: $\quad\theta_0^{(\tau)} \leftarrow \theta_0$
6: $\quad\theta^{(\tau)} \leftarrow \mathrm{PPO}\left(\theta_0^{(\tau)}, \Delta^{(\tau)}\right)$
7: $\quad$**for** $g \in \mathcal{G}$ **do**
8: $\quad\quad\theta_{0,g} \leftarrow \arg \max_{v \in \{\theta_{0,g},\, \theta_g^{(\tau)}\}} \|v\|_1$
9: $\quad$**end for**
10: **end for**

Although more directly tied to the local objective, these scores incur additional costs under a strict interaction budget. The first-order criterion requires estimating $g^{(\tau)}$ from current-task rollouts before initialization, while the second-order criterion further requires curvature estimation; both can be noisy with limited samples. Moreover, both of them require access to historical neuron groups, increasing storage and selection overhead as more tasks are observed.

The proposed criterion instead uses the Manhattan norm as an interaction-free proxy for saliency accumulated during previous optimization. A historical neuron group can be written schematically as

$$\theta_{i,g} = \theta_g^{\mathrm{init}} + \sum_{k=1}^{K_i} \eta_k \widehat{g}_{i,g}^{(k)}, \tag{13}$$

where consistently reinforced updates can lead to a larger final magnitude. The Manhattan norm is not an exact approximation of the first- or second-order score, but a surrogate that avoids additional rollout estimation and historical model storage, making it suitable for budgeted CRL.

4.4 Overall Procedure

Based on the neuron-wise Manhattan-norm criterion above, we term the proposed method *Neuron-wise mAnhattan-norm Maximization Initialization* (NAMI), whose overall procedure is summarized in Algorithm 1. Although Eq. (9) allows each neuron group to select from all historical policies, explicitly storing all previous models would incur growing overhead. We therefore maintain a global network θ_0 that recursively stores the neuron-wise winners observed so far.

For the first task, the policy is randomly initialized and trained, after which the trained policy is used to initialize the global network θ_0 (Lines 1–3):

$$\theta^{(1)} \leftarrow \mathrm{PPO}\left(\theta_0^{(1)}, \Delta^{(1)}\right), \qquad \theta_0 \leftarrow \theta^{(1)}. \tag{14}$$

For each subsequent task $\tau > 1$, the current policy is initialized directly from the global network θ_0 and trained under the interaction budget (Lines 5–6):

$$\theta_0^{(\tau)} \leftarrow \theta_0, \qquad \theta^{(\tau)} \leftarrow \mathrm{PPO}\left(\theta_0^{(\tau)}, \Delta^{(\tau)}\right). \tag{15}$$

After training, the global network θ_0 is updated neuron-wise by comparing the Manhattan norms of the stored and newly learned neuron groups (Lines 7–9):

$$\theta_{0,g} \leftarrow \arg \max_{v \in \{\theta_{0,g},\, \theta_g^{(\tau)}\}} \|v\|_1, \qquad \forall g \in \mathcal{G}. \tag{16}$$

This recursive update retains the largest-Manhattan-norm neuron block among all policies observed so far, allowing the global network θ_0 to progressively consolidate salient and reusable knowledge across tasks. As a result, θ_0 serves as a compact initialization for future tasks without requiring storage of all historical policy networks. The overall process is repeated for each subsequent task until the end of the task sequence (Lines 4–10).

Table 1: Overall performance comparison across `Freeway`, `SpaceInvaders`, and `CrossGame`. Best results are in **bold**, and second-best results are underlined.

Method	Freeway		SpaceInvaders		CrossGame	
	FP (↑)	FWT (↑)	FP (↑)	FWT (↑)	FP (↑)	FWT (↑)
FS	0.1891±0.3473	0.0000±0.0000	0.6163±0.3493	0.0000±0.0000	0.9625±0.0808	0.0000±0.0000
FT-N (Wołczyk et al., 2021)	0.9287±0.0669	0.7544±0.0943	0.9971±0.0039	0.7984±0.1772	0.9842±0.0414	0.1356±0.3737
ProgNet (Rusu et al., 2016)	0.4245±0.3981	0.2429±0.2735	0.7159±0.2501	0.0405±0.1106	0.9908±0.0174	0.0044±0.2991
PackNet (Mallya & Lazebnik, 2018)	0.2664±0.3839	0.0420±0.3174	0.2336±0.3244	-0.1508±0.1592	0.3944±0.4145	-0.5487±0.3016
MaskNet (Ben-Iwhiwhu et al., 2023)	0.1250±0.3280	-0.0981±0.2418	0.0991±0.2974	-0.3567±0.3489	0.1252±0.3307	-2.3045±2.2904
CReLUs (Abbas et al., 2023)	0.9099±0.0702	0.7284±0.0979	0.9968±0.0060	0.7986±0.1944	0.9981±0.0043	0.1498±0.3783
CbpNet (Dohare et al., 2024)	0.8939±0.0823	0.7060±0.1020	0.9954±0.0107	0.7624±0.2089	0.9913±0.0230	0.0576±0.4388
TRAC (Muppidi et al., 2024)	0.1241±0.3283	-0.0994±0.2413	0.2559±0.3467	-0.1605±0.1871	0.3956±0.4608	-1.5133±2.0087
CompoNet (Malagón et al., 2024)	0.9329±0.0570	0.7528±0.0795	0.9974±0.0036	0.6995±0.2519	0.9720±0.0702	0.0917±0.3160
C-CHAIN (Tang et al., 2025)	0.9113±0.0840	0.7331±0.1044	0.9977±0.0034	0.7962±0.1862	0.9920±0.0212	0.1548±0.4338
CKA-RL (Hu et al., 2025)	0.9137±0.0724	0.7316±0.0928	0.9971±0.0049	0.7568±0.2049	**0.9996±0.0008**	0.2369±0.3553
ReGraMa (Liu et al., 2026)	0.9264±0.0636	0.7507±0.0891	0.9972±0.0051	0.7755±0.2038	0.9982±0.0049	0.2514±0.3219
NAMI (Ours)	**0.9382±0.0560**	**0.7612±0.0871**	**0.9978±0.0039**	**0.8087±0.1718**	**0.9996±0.0012**	**0.2825±0.3993**

5 Experiments

5.1 Experimental Setup

Benchmarks. We evaluate NAMI on Atari tasks constructed from the Arcade Learning Environment (ALE) (Machado et al., 2018). We consider two types of sequences: cross-mode and cross-game settings. In the cross-mode setting, tasks are different modes of the same Atari game: `Freeway` with 8 modes, `SpaceInvaders` with 10 modes, `Hero` with 4 modes, and `WordZapper` with 18 modes. In the cross-game setting, we construct an 8-task sequence consisting of `Pacman`, `MsPacman`, `Amidar`, `Entombed`, `Berzerk`, `Alien`, `Tutankham`, `BankHeist`.

Metrics. We report average performance, final performance, and forward transfer as metrics for CRL. Let $p_\tau(t) \in [0, 1]$ denote the success rate on task τ at training step t. The total number of training steps is $T = N\Delta$ for a sequence of N tasks, each trained for Δ steps. We define

$$AP(t) = \frac{1}{t}\sum_{j=1}^{t} FP(j), \qquad FP(t) = \frac{1}{N}\sum_{i=1}^{N} p_i(t), \tag{17}$$

where $AP(T)$ and $FP(T)$ measure the average performance and final performance, respectively.

For forward transfer, we compare the learning curve against the model trained from scratch on task τ. Let $p^b_\tau(t) \in [0, 1]$ denote the reference success rate, the forward transfer on task τ is then defined as

$$FT_i = \frac{A_i - A^b_i}{1 - A^b_i}, \quad A_\tau = \frac{1}{\Delta}\int_{(\tau-1)\Delta}^{\tau\Delta} p_\tau(t)\, dt, \quad A^b_\tau = \frac{1}{\Delta}\int_{0}^{\Delta} p^b_\tau(t)\, dt. \tag{18}$$

Finally, we report the final forward transfer across all tasks as

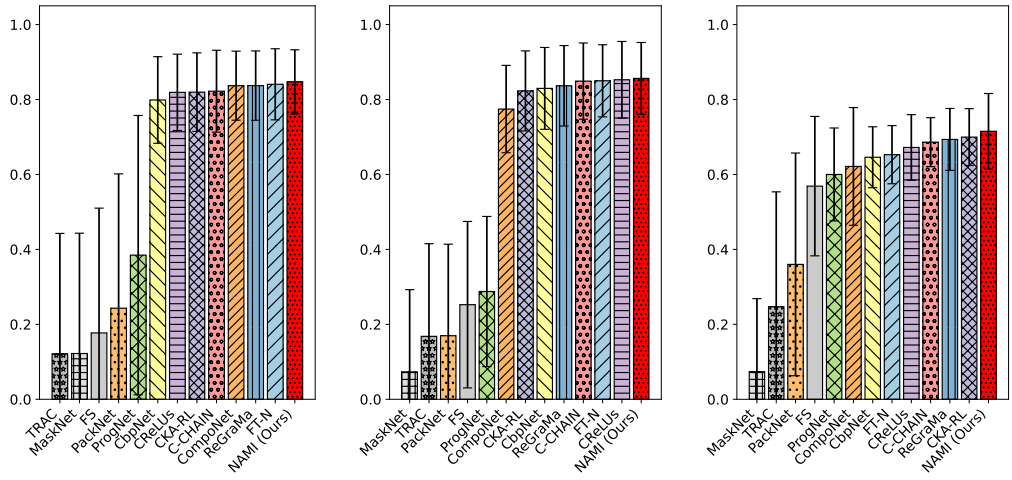
$$FWT = \frac{1}{N}\sum_{i=1}^{N} FT_i, \tag{19}$$

where $FWT > 0$ indicates positive transfer (faster learning than from scratch) and vice versa.

Implementation Details. For Atari, 210×160 RGB observations are encoded by a CNN and passed to actor/critic MLP heads. Each task is trained for $\Delta = 1$M environment steps using PPO (Schulman et al., 2017). Following prior work (Wolczyk et al., 2022; Malagón et al., 2024; Hu et al., 2025), continual learning is applied only to the actor, while the critic is reinitialized for each task. The actor uses a 2-layer MLP with 512 hidden units. We use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\gamma = 0.99$, learning rate 2.5×10^{-4}, and batch size 1024. Results are averaged over 10 seeds (From 0 to 9).

5.2 Overall Performance

Final performance evaluation. Table 1 compares the final performance (FP) of all methods. NAMI achieves the highest FP on all three benchmarks, obtaining 0.9382 on `Freeway`, 0.9978 on `SpaceInvaders`, and 0.9996 on `CrossGame`. Even on `SpaceInvaders` and `CrossGame`, where several baselines approach saturated performance, NAMI remains competitive at the top. This means that NAMI improves both learning efficiency and the quality of the final learned policies.



(a) Freeway (b) SpaceInvaders (c) Cross Game

Figure 4: Average performance (↑) comparison across all tasks.

Table 2: Overall performance comparison across `Hero` and `WordZapper`.

Method	**Hero**		**WordZapper**	
	FP (↑)	FWT (↑)	FP (↑)	FWT (↑)
FS	0.8020±0.40	0.0000±0.00	0.6126±0.31	0.0000±0.00
FT-N (Wołczyk et al., 2021)	0.9249±0.13	0.2721±0.19	0.6086±0.26	0.0953±0.15
CKA-RL (Hu et al., 2025)	**0.9896±0.02**	0.3368±0.14	0.5716±0.24	0.0673±0.13
NAMI (Ours)	0.9857±0.02	**0.3475±0.16**	**0.6446±0.25**	**0.1017±0.15**

Forward transfer evaluation. Table 1 also reports forward transfer (FWT), which measures how effectively prior knowledge accelerates learning on future tasks. NAMI achieves the highest FWT on all benchmarks. The gain is particularly pronounced on `CrossGame`, where NAMI substantially outperforms ReGraMa. These results suggest that neuron-wise knowledge reuse provides a more effective initialization by selectively combining transferable components from previous tasks.

Overall evaluation. Figure 4 shows the average performance (AP) comparison across all tasks, providing an overall evaluation of how well each method performs throughout the learning process. NAMI achieves the best AP on all three benchmarks, with 0.8470 on `Freeway`, 0.8563 on `SpaceInvaders`, and 0.7152 on `CrossGame`. These results show that NAMI consistently improves performance across the task stream, suggesting that neuron-wise knowledge transfer provides a robust mechanism for overall performance gains.

Evaluation on sequence length. As demonstrated in Table 2, we further evaluate NAMI on two more environments: `Hero` with fewer modes and `WordZapper` with a longer task sequence and more frequent distribution shifts. On `Hero`, NAMI is comparable to the strongest baseline, while CKA-RL achieves slightly higher FP at the cost of maintaining five knowledge vectors. In contrast, NAMI exhibits a clearer advantage on the longer `WordZapper` sequence, where knowledge must be transferred across more tasks and distribution shifts. These results suggest that NAMI benefits more from longer task sequences by enabling efficient knowledge accumulation and reuse across tasks.

5.3 Further Study

Ablation Study. Table 3 reports the ablation results on `Freeway`. NAMI-Param-No-Abs slightly underperforms FT-N, indicating that raw-value maximization alone is insufficient. NAMI-Param improves AP, FP, and FWT by selecting parameters based on their absolute magnitudes, while extending the selection to complete neuron groups further improves all three metrics, with NAMI achieving the best overall performance. The NAMI-FO and NAMI-SO variants do not outperform NAMI, suggesting that explicitly incorporating first- and second-order information provides no clear

Table 3: Ablation study of NAMI on `Freeway`. Best results are in **bold**.

Method	Performance			Components					
	AP (↑)	FP (↑)	FWT (↑)	Maximization	Absolute	Neuron	First	Second	Reset
FT-N	0.7036±0.1150	0.7997±0.1058	0.6477±0.1176						
NAMI-Param-No-Abs	0.7027±0.1347	0.7942±0.1224	0.6471±0.1380	✓					
NAMI-Param	0.7096±0.1261	0.8022±0.1104	0.6544±0.1300	✓	✓				
NAMI	**0.7167±0.1163**	**0.8109±0.0990**	**0.6619±0.1208**	✓	✓	✓			
NAMI-FO	0.7094±0.1258	0.8028±0.1124	0.6541±0.1299	✓	✓	✓	✓		
NAMI-SO	0.6885±0.1302	0.7789±0.1138	0.6314±0.1322	✓	✓	✓	✓	✓	
NAMI-Reset	0.6528±0.0860	0.7552±0.0978	0.5903±0.0762	✓	✓	✓			✓

advantage under the limited interaction. Finally, NAMI-Reset substantially degrades performance, further supporting the importance of preserving neurons with high magnitudes for forward transfer.

Evaluation on continuous control. We further evaluate NAMI on Meta-World (Yu et al., 2020) using the first five tasks of CW10 (Wołczyk et al., 2021): `Hammer`, `Push-Wall`, `Faucet-Close`, `Push-Back`, and `Stick-Pull`. Different from Atari, Meta-World uses a Gaussian continuous-action policy with separate mean and standard-deviation heads. As shown in Table 4, NAMI achieves the highest FP and FWT among the compared methods. These results provide additional evidence that neuron-wise knowledge selection is not specific to discrete-action Atari environments and can also generalize to continuous-control policies.

Table 4: Performance comparison on Meta-World. Best results are in **bold**.

Method	FP (↑)	FWT (↑)
Fine-tuning	0.2945±0.4558	0.0000±0.0000
CKA-RL	0.2912±0.4543	0.0304±0.0571
NAMI	**0.3118±0.4632**	**0.0318±0.0411**

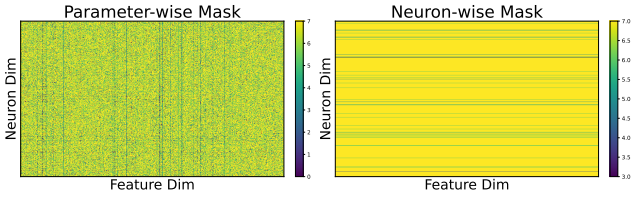


Figure 5: The heatmap of the mask matrix in two ways.

Selection Visualization. We use a mask matrix to record the source task of each hidden-layer parameter. Figure 5 compares parameter-wise and neuron-wise masks on `Freeway`, where darker colors denote earlier tasks. Parameter-wise selection produces fragmented masks across many tasks, whereas neuron-wise selection yields a more structured pattern dominated by recent tasks. This suggests that neuron-wise selection preserves coherent transferable structure while reducing excessive cross-task mixing.

Complexity Analysis. Table 5 compares the computational complexity of different methods, where P is the number of parameters in one policy network and τ the number of observed tasks. NAMI retains the $O(P)$ transfer, training, and inference complexity of FT-N while enabling richer knowledge reuse. In contrast, CompoNet incurs $O(\tau^2 P)$ training and inference costs due to sequential policy composition, while CKA-RL requires $O(\tau P)$ transfer and training through model-level combination. Using neuron-wise knowledge transfer, NAMI avoids complexity growth with the number of tasks and preserves the efficiency of FT-N.

Table 5: Complexity comparison.

Method	Transfer	Training	Inference
FT-N	$O(P)$	$O(P)$	$O(P)$
CompoNet	$O(\tau P)$	$O(\tau^2 P)$	$O(\tau^2 P)$
CKA-RL	$O(\tau P)$	$O(\tau P)$	$O(P)$
NAMI	$O(P)$	$O(P)$	$O(P)$

6 Conclusion

In this work, we present NAMI, a neuron-wise initialization strategy for efficient forward transfer in CRL. Motivated by the limitation of coarse-grained transfer methods, NAMI reformulates fine-grained forward transfer as a combinatorial optimization problem in the parameter space and uses the Manhattan norm as an efficient surrogate for selecting reusable neuron groups from the previous policy networks. This design enables the new policy network to inherit informative components from multiple past tasks while remaining a single compact network with constant inference cost. Experiments on ALE benchmarks show that NAMI consistently improves forward transfer and final performance compared with representative architecture-based methods.

AI USE STATEMENT

Generative AI tools were used solely to assist with manuscript writing and editing, including improving language, clarity, and readability. They were not used for research problem formulation, method development, experimental design, code generation, data analysis, or result interpretation. All AI-assisted text was carefully reviewed and revised by the authors, who take full responsibility for the final content of the manuscript.

ETHICS STATEMENT

This work studies continual reinforcement learning algorithms using standard simulated benchmark environments. It does not involve human subjects, personally identifiable information, or sensitive data. The experiments are conducted entirely in simulated environments, and we do not identify specific ethical risks. The authors have followed the ICLR Code of Ethics throughout the research and reporting process.

REPRODUCIBILITY STATEMENT

We provide the formulation and complete procedure of NAMI in Section 4, including the overall algorithm in Algorithm 1. The benchmark construction, evaluation metrics, network architecture, training budgets, optimization settings, and random seeds are described in Section 5.1. Additional ablation studies and analyses are reported in Section 5.3. These details are provided to facilitate reproduction of the experimental results.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado P van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36:50377–50407, 2023.

Shaghaf Alzorba, Christian Günther, Nicolae Popovici, and Christiane Tammer. A new algorithm for solving planar multiobjective location problems involving the manhattan norm. *European Journal of Operational Research*, 258(1):35–46, 2017.

Eseoghene Ben-Iwhiwhu, Saptarshi Nath, Praveen K Pilly, Soheil Kolouri, and Andrea Soltoggio. Lifelong reinforcement learning with modulating masks. *Trans. Mach. Learn. Res.*, 2023.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Haotian Fu, Yixiang Sun, Michael Littman, and George Konidaris. Knowledge retention in continual model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 17832–17851. PMLR, 2025.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pp. 11321–11339. PMLR, 2023.

Jinwu Hu, Zihao Lian, Zhiquan Wen, Chenghao Li, Guohao Chen, Xutao Wen, Bin Xiao, and Mingkui Tan. Continual knowledge adaptation for reinforcement learning. *arXiv preprint arXiv:2510.19314*, 2025.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Böhmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *9th International Conference on Learning Representations*, 2021.

Samuel Kessler, Jack Parker-Holder, Philip Ball, Stefan Zohren, and Stephen J Roberts. Unclear: A straightforward method for continual reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Qiyang Li, Yuexiang Zhai, Yi Ma, and Sergey Levine. Understanding the complexity gains of single-task rl with a curriculum. In *International Conference on Machine Learning*, pp. 20412–20451. PMLR, 2023.

Huiwei Lin, Shanshan Feng, Baoquan Zhang, Hongliang Qiao, Xutao Li, and Yunming Ye. Uer: A heuristic bias addressing approach for online continual learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 96–104, 2023a.

Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24246–24255, 2023b.

Jiashun Liu, Zihao Wu, Johan Obando Ceron, Pablo Samuel Castro, Aaron Courville, and Ling Pan. Measure gradients, not activations! enhancing neuronal activity in deep reinforcement learning. *Advances in Neural Information Processing Systems*, 38:79546–79571, 2026.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Mikel Malagón, Josu Ceberio, and Jose A Lozano. Self-composing policies for scalable continual reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 34432–34460, 2024.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Aneesh Muppidi, Zhiyu Zhang, and Heng Yang. Fast trac: A parameter-free optimizer for lifelong reinforcement learning. *Advances in Neural Information Processing Systems*, 37:51169–51195, 2024.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ke Sun, Hongming Zhang, Jun Jin, Chao Gao, Xi Chen, Wulong Liu, and Linglong Kong. Principled fast and meta knowledge learners for continual reinforcement learning. In *ICLR*, 2025.

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

Hongyao Tang, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn. *arXiv preprint arXiv:2506.00592*, 2025.

Fengqiang Wan and Yang Yang. Probabilistic group mask guided discrete optimization for incremental learning. In *Forty-second International Conference on Machine Learning*, 2025.

Maciej Wołczyk, Michał Zajkac, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Maciej Wolczyk, Michał Zajkac, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Disentangling transfer in continual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:6304–6317, 2022.

Annie Xie, Logan Mondal Bhamidipaty, Evan Zheran Liu, Joey Hong, Sergey Levine, and Chelsea Finn. Learning to explore in pomdps with informational rewards. In *Forty-first International Conference on Machine Learning*, 2024.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.