Paper Report

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: REINFORCEMENT L EARNING WITH UNSUPERVISED AUXILIARY TASKS
Date: 18/11/2024

---

1 Introduction

Reinforcement learning (RL) has become a cornerstone of modern artificial intelligence, with deep reinforcement learning (DRL) agents demonstrating exceptional capabilities across a range of complex tasks. These agents achieve state-of-the-art performance by directly maximizing cumulative extrinsic rewards provided by the environment. However, many real-world environments present challenges where extrinsic rewards are sparse or infrequent, limiting the effectiveness of traditional DRL approaches.

In such scenarios, the agent's ability to learn robust and generalizable representations is critical. Beyond the sparse extrinsic rewards, environments contain a plethora of implicit training signals that, if leveraged effectively, can significantly accelerate learning and improve performance. This paper introduces a novel approach to reinforcement learning that augments traditional methods by incorporating multiple auxiliary tasks. These auxiliary tasks, akin to pseudo-rewards, guide the agent to learn diverse and meaningful features of its environment. Crucially, this approach extends the agent's ability to learn even in the absence of explicit extrinsic rewards, drawing inspiration from the principles of unsupervised learning.

To address the challenge of aligning representation learning with task-specific goals, a novel mechanism is introduced that enables the agent to dynamically focus on the most relevant aspects of its task. This mechanism allows the representation to adapt efficiently to changes in the task or environment, enhancing both the speed and robustness of learning.

The proposed agent, termed the UNsupervised REinforcement and Auxiliary Learning (UNREAL) agent, combines the advantages of Asynchronous Advantage Actor-Critic (A3C) reinforcement learning with auxiliary tasks designed to maximize pseudo-rewards. The UNREAL agent achieves significant improvements in performance across multiple domains. For example, in the Atari domain, it averages 880% of expert human performance, surpassing prior benchmarks. In the challenging 3D Labyrinth environment, the UNREAL agent demonstrates an average learning speedup of 10× compared to baseline approaches, achieving 87% of expert human performance.

This paper highlights how the integration of auxiliary tasks and adaptive representations can unlock new levels of efficiency and effectiveness in reinforcement learning, paving the way for agents capable of thriving in highly complex and dynamic environments.

2 Literature Survey

Reinforcement learning (RL) has seen significant advancements, with various architectures exploring different facets of learning and representation. This section highlights key contributions in the field, placing the proposed UNREAL agent in context.

Temporal Abstractions and Pseudo-Rewards

One line of work has focused on temporal abstractions to facilitate high-level planning. Notable examples include the Options Framework and pseudo-reward maximization approaches

such as those by Konidaris and Barreto. These methods aim to define intermediate goals to improve the scalability and efficiency of RL agents. However, these frameworks primarily utilize pseudo-rewards for planning and do not integrate them as auxiliary objectives for representation learning.

The Horde architecture is another example that learns multiple value functions corresponding to distinct pseudo-rewards. Although effective, this architecture isolates value functions with distinct weights, lacking the shared representation that enables multi-task learning.

Representation Learning in RL

Representation learning has emerged as a vital component of modern RL systems. Universal Value Function Approximators (UVFAs) leverage embeddings of state features and pseudo-reward functions to encode continuous sets of value functions. While this approach supports transfer learning across tasks, it primarily focuses on the embedding structure and does not actively incorporate auxiliary objectives.

Successor representations provide a mechanism for transferring value functions between tasks by factoring state features and rewards. Kulkarni et al. extended this concept to scale rewards dynamically. While promising, these methods are limited to value-based transfer and do not generalize to auxiliary tasks for representation learning.

Auxiliary Predictions in 3D Environments

Recent research has emphasized auxiliary prediction tasks to enhance learning in complex 3D environments. Lample and Chaplot demonstrated the benefits of predicting game-specific features (e.g., the presence of an enemy). Mirowski et al. explored auxiliary depth predictions to improve navigation capabilities. These studies underscore the potential of auxiliary tasks but are often task-specific and do not leverage unsupervised learning principles to adapt representations.

Model-Based RL and Environmental Dynamics

Another line of research involves learning environment models to support planning. For instance, Schmidhuber explored intrinsic motivation for unsupervised learning in RL, while Oh et al. investigated video frame prediction as an auxiliary task. Despite their success in model-based scenarios, these methods have yet to demonstrate significant improvements in rich visual environments.

UNREAL Agent and Novel Contributions

The UNREAL agent distinguishes itself by combining state-of-the-art on-policy methods such as Asynchronous Advantage Actor-Critic (A3C) with auxiliary tasks for representation learning. Unlike prior work, it integrates control and reward-based auxiliary objectives to develop adaptable and efficient representations. The agent leverages auxiliary control tasks (e.g., pixel control) and reward prediction tasks to enhance both data efficiency and learning robustness, achieving superior performance in visual domains like Atari and Labyrinth.

By addressing gaps in prior research, the UNREAL agent demonstrates how auxiliary tasks can unlock the potential of unsupervised reinforcement learning, enabling significant improvements in speed, scalability, and generalization.

3 Methodology

The UNsupervised REinforcement and Auxiliary Learning (UNREAL) agent builds upon the Asynchronous Advantage Actor-Critic (A3C) framework while integrating auxiliary tasks to

enhance learning efficiency, robustness, and representation quality. This section details the architecture, auxiliary tasks, and loss function formulation of the UNREAL agent.

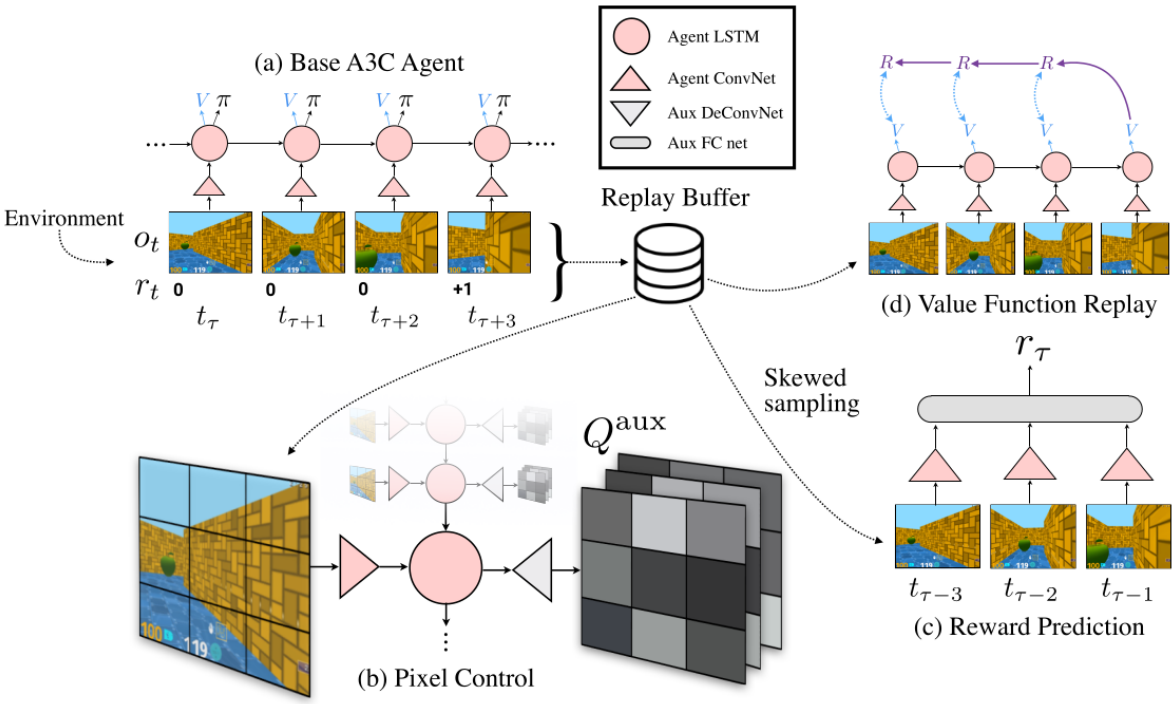

Figure 1: Overview of the UNREAL agent. (a) The base agent is a CNN-LSTM agent trained on-policy with the A3C loss (Mnih et al., 2016). Observations, rewards, and actions are stored in a small replay buffer which encapsulates a short history of agent experience. This experience is used by auxiliary learning tasks. (b) Pixel Control – auxiliary policies Qaux are trained to maximise change in pixel intensity of different regions of the input. The agent CNN and LSTM are used for this task along with an auxiliary deconvolution network. This auxiliary control task requires the agent to learn how to control the environment. (c) Reward Prediction – given three recent frames, the network must predict the reward that will be obtained in the next unobserved timestep. This task network uses instances of the agent CNN, and is trained on reward biased sequences to remove the perceptual sparsity of rewards. (d) Value Function Replay – further training of the value function using the agent network is performed to promote faster value iteration.

Base Agent: A3C Framework

The UNREAL agent uses the A3C algorithm as its backbone. A3C trains agents asynchronously by interacting with multiple environment instances in parallel, stabilizing learning and improving efficiency. The agent is modeled using a convolutional neural network (CNN) for feature extraction, followed by a Long Short-Term Memory (LSTM) module to capture temporal dependencies in partially observable environments.

The A3C loss function, L_{A3C}, combines policy optimization, value estimation, and entropy regularization:

$$L_{\text{A3C}} = L_{\text{VR}} + L_{\pi} - \alpha H(\pi),$$

where L_{VR} is the value regression loss, L_{π} is the policy gradient loss, and $H(\pi)$ is the entropy of the policy, promoting exploration.

Auxiliary Tasks for Representation Learning

The UNREAL agent augments the A3C base with auxiliary tasks designed to improve the agent's representation of the environment. These tasks include:

1. Pixel Control

Pixel control aims to maximize changes in pixel intensities across spatial regions of the input image. The input is divided into a grid, and policies are trained to maximize pixel changes in each grid cell. This task is formulated as an n-step Q-learning problem, with the auxiliary Q-learning loss:

$$L_{\text{PC}} = \mathbb{E}\left[\left(R^{(c)}_{t:t+n} + \gamma^n \max_{a'} Q_{\text{aux}}(s', a'; \theta^-) - Q_{\text{aux}}(s, a; \theta)\right)^2\right],$$

where Q_{aux} represents the auxiliary action-value estimates.

2. Reward Prediction

This task predicts immediate rewards based on a sequence of past observations. The agent processes three consecutive input frames using the shared CNN, followed by a feedforward network for classification into three reward classes: positive, negative, or zero. The prediction loss is defined as:

$$L_{\text{RP}} = -\sum_{c \in \{+,-,0\}} p_c \log \hat{p}_c,$$

where p_c and $\hat{p}_c$ are the true and predicted probabilities of each reward class, respectively.

3. Value Function Replay

To stabilize and accelerate value function learning, the agent replays past experience from a buffer and applies additional value regression updates. The replay loss is given by:

$$L_{\text{VR}} = \mathbb{E}\left[\left(R_{t:t+n} + \gamma^n V(s_{t+n+1}; \theta^-) - V(s_t; \theta)\right)^2\right].$$

Unified Loss Function

The total loss function for the UNREAL agent combines the A3C loss with the auxiliary task losses, weighted by their respective coefficients:

$$L_{\text{UNREAL}} = L_{\text{A3C}} + \lambda_{\text{PC}} L_{\text{PC}} + \lambda_{\text{RP}} L_{\text{RP}} + \lambda_{\text{VR}} L_{\text{VR}},$$

where $\lambda_{\text{PC}}, \lambda_{\text{RP}}, \lambda_{\text{VR}}$ are hyperparameters controlling the influence of each auxiliary task.

Experience Replay and Prioritized Sampling

The agent employs a replay buffer to store recent experience, enabling off-policy updates for auxiliary tasks. For reward prediction, the replay buffer is skewed to equally represent rewarding and non-rewarding events, mitigating the sparsity of rewards in the environment. In contrast, value function replay uses uniform sampling without bias.

Integration and Training

The UNREAL algorithm trains the agent by alternating on-policy updates for the A3C loss and off-policy updates for auxiliary task losses. Auxiliary tasks are performed every 20 environment steps, aligning with updates to the base A3C agent. The shared CNN and LSTM layers ensure that auxiliary tasks directly shape the representation used for policy learning.

This methodology enables the UNREAL agent to achieve superior performance by leveraging auxiliary objectives to focus representation learning on task-relevant features.

4 Experimentation Setup

The experiments were conducted using an A3C CNN-LSTM agent as the baseline, with the proposed UNREAL agent and its variants augmented by auxiliary tasks. The key details are as follows:

- **Input:** The agent receives an 84×84 RGB image as input at each timestep.
- **Base Architecture:**
  - Two convolutional layers:
    - First layer: 16 filters of size 8×8 with stride 4.
    - Second layer: 32 filters of size 4×4 with stride 2.
  - A fully connected layer with 256 units.
  - An LSTM with 256 cells, which processes the CNN-encoded observation concatenated with the previous action and reward.
- **Training Setup:**
  - **Step Returns:** The agents were trained on-policy with 20-step returns.
  - **Replay Buffer:** Stores the most recent $2k$ observations, actions, and rewards taken by the base agent.
- **Action Space:**
  - Atari: Game-dependent (3 to 18 discrete actions).
  - Labyrinth: Fixed set of 17 discrete actions.
- **Frame Processing:** The Labyrinth environment runs at 60 frames-per-second, with actions repeated for 4 frames.

Auxiliary Tasks

- **Pixel Control:**
  - Operates on the central 80×80 crop of the inputs.
  - The cropped region is subdivided into a 20×20 grid of 4×4 cells.
  - Rewards correspond to average absolute pixel changes per cell.
- **Feature Control:**
  - Targets the second hidden layer, a $32 \times 9 \times 9$ spatial feature map.
  - Uses a deconvolutional network for processing.
- **Reward Prediction:**

  - Trains a feedforward network on sequences of 3 observations to predict rewards.
  - Classifies rewards as positive, negative, or zero using a softmax output.

- **Value Function Replay:**
  - Operates on sequences of length 20 with regression loss.

Optimization

- Agents were optimized using 32 asynchronous threads and a shared RMSProp optimizer.
- Learning rates were sampled log-uniformly between 0.0001 and 0.005.
- Entropy costs were sampled log-uniformly between 0.0005 and 0.01.

Evaluation

- Labyrinth: 13 levels tested on various navigation and control tasks.
- Atari: 57 games tested using human-normalized scores.

5 Conclusion

The integration of auxiliary control tasks and reward prediction mechanisms into a deep reinforcement learning framework has demonstrated remarkable improvements in both the efficiency of data usage and the robustness of the learning process to variations in hyperparameter settings. These enhancements are particularly noteworthy in environments that pose significant challenges, such as the 3D Labyrinth levels, where the UNREAL (UNsupervised REinforcement and Auxiliary Learning) architecture significantly outperformed existing state-of-the-art methods.

The UNREAL agent achieved an average score that exceeded 87% of expert human performance on these 3D Labyrinth tasks. This result represents more than a twofold improvement compared to previous best-performing methods. These improvements are attributed to the incorporation of auxiliary tasks, which enable the agent to learn more effectively by leveraging additional signals within the environment. By doing so, the agent develops a richer and more adaptive representation of the state space, facilitating faster and more stable learning.

Beyond the Labyrinth environment, the benefits of the UNREAL architecture were also evident across a suite of 57 Atari games, a benchmark widely used to evaluate reinforcement learning algorithms. In this domain, the UNREAL agent exhibited a significant enhancement in learning speed, allowing it to achieve high levels of performance with fewer training steps. Furthermore, the architecture proved to be robust against variations in critical hyperparameters, such as learning rate and entropy cost, ensuring reliable performance across a diverse range of configurations.

In summary, the UNREAL architecture not only sets a new standard for performance in challenging environments but also highlights the importance of leveraging auxiliary tasks to enrich the learning process. This approach ensures that reinforcement learning agents are not only more efficient in their training but also capable of achieving superior levels of performance across a variety of tasks and environments.

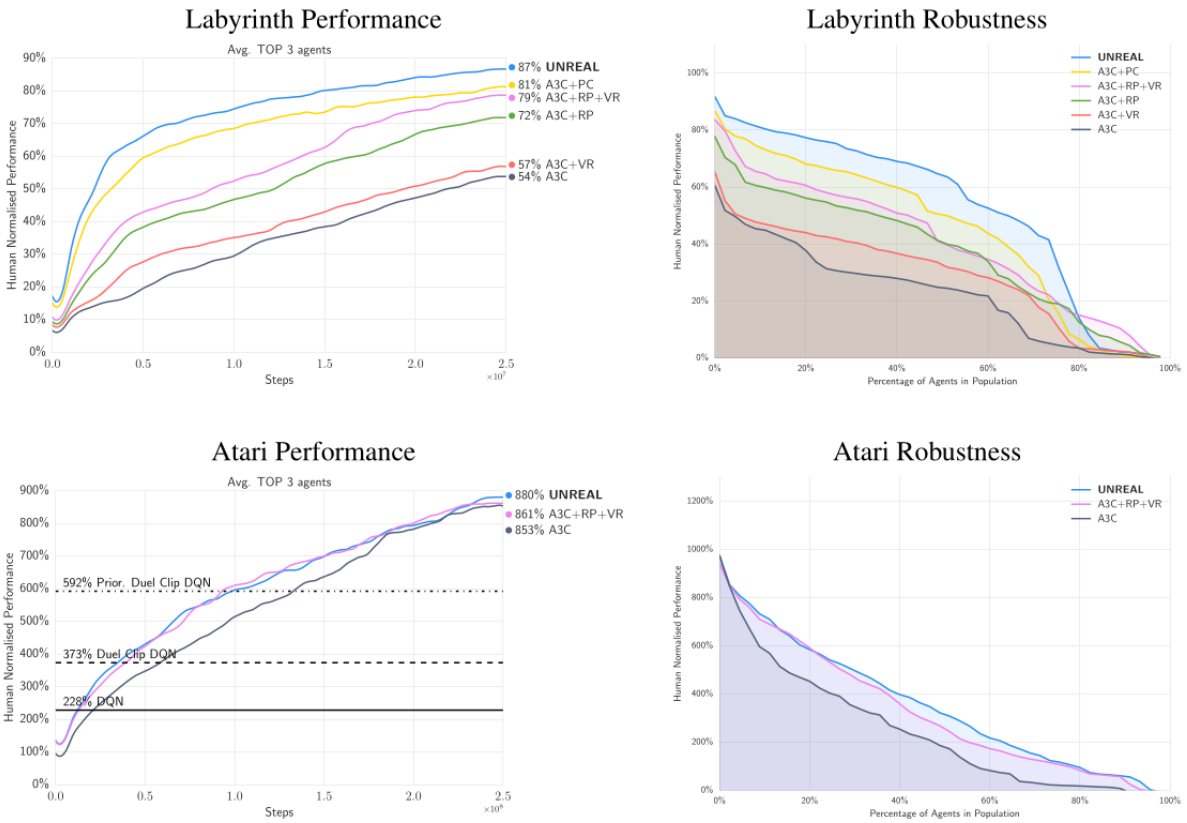

Figure 2: An overview of performance averaged across all levels on Labyrinth (Top) and Atari (Bottom). In the ablated versions RP is reward prediction, VR is value function replay, and PC is pixel control, with the UNREAL agent being the combination of all. Left: The mean human-normalised performance over last 100 episodes of the top-3 jobs at every point in training. We achieve an average of 87% human normalised score, with every element of the agent improving upon the 54% human-normalised score of vanilla A3C. Right: The final human-normalised score of every job in our hyperparameter sweep, sorted by score. On both Labyrinth and Atari, the UNREAL agent increases the robustness to the hyperparameters (namely learning rate and entropy cost).

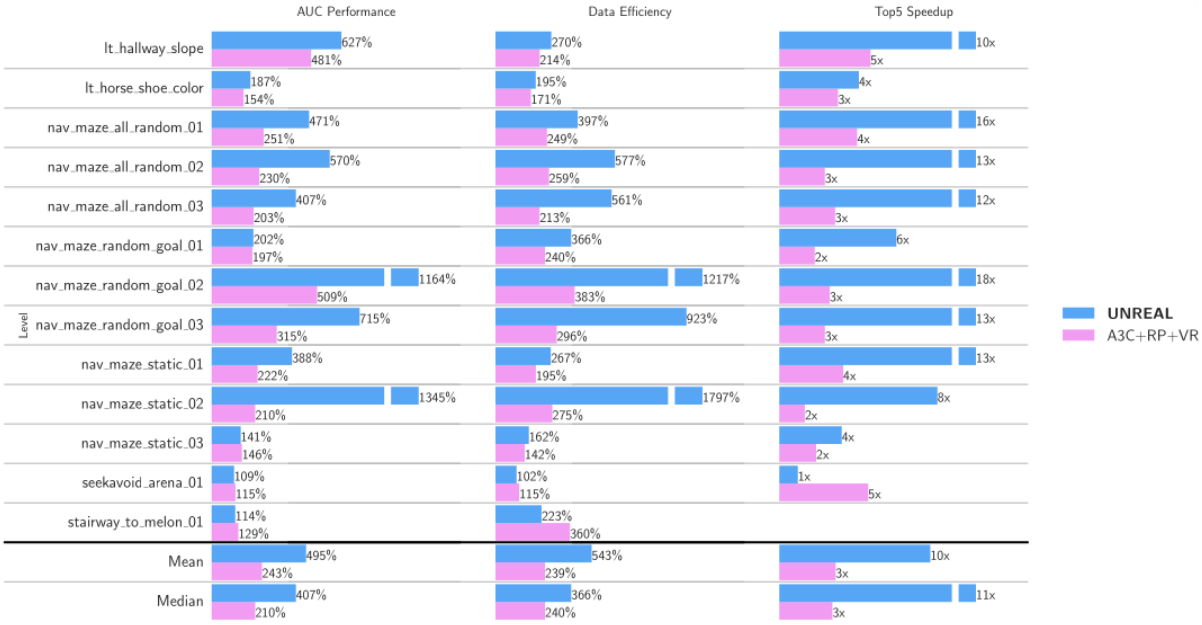

Figure 3: A breakdown of the improvement over A3C due to our auxiliary tasks for each level on Labyrinth. The values for A3C+RP+VR (reward prediction and value function replay) and UNREAL (reward prediction, value function replay and pixel control) are normalised by the A3C value. AUC Performance gives the robust- ness to hyperparameters (area under the robustness curve Figure 3 Right). Data Efficiency is area under the mean learning curve for the top-5 jobs, and Top5 Speedup is the speedup for the mean of the top-5 jobs to reach the maximum top-5 mean score set by A3C. Speedup is not defined for stairway to melon as A3C did not learn throughout training.

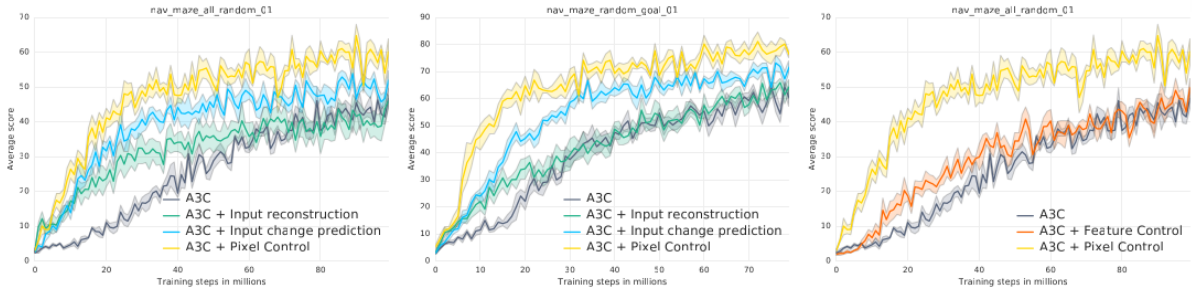

Figure 4: Comparison of various forms of self-supervised learning on random maze navigation. Adding an input reconstruction loss to the objective leads to faster learning compared to an A3C baseline. Predicting changes in the inputs works better than simple image reconstruction. Learning to control changes leads to the best results.