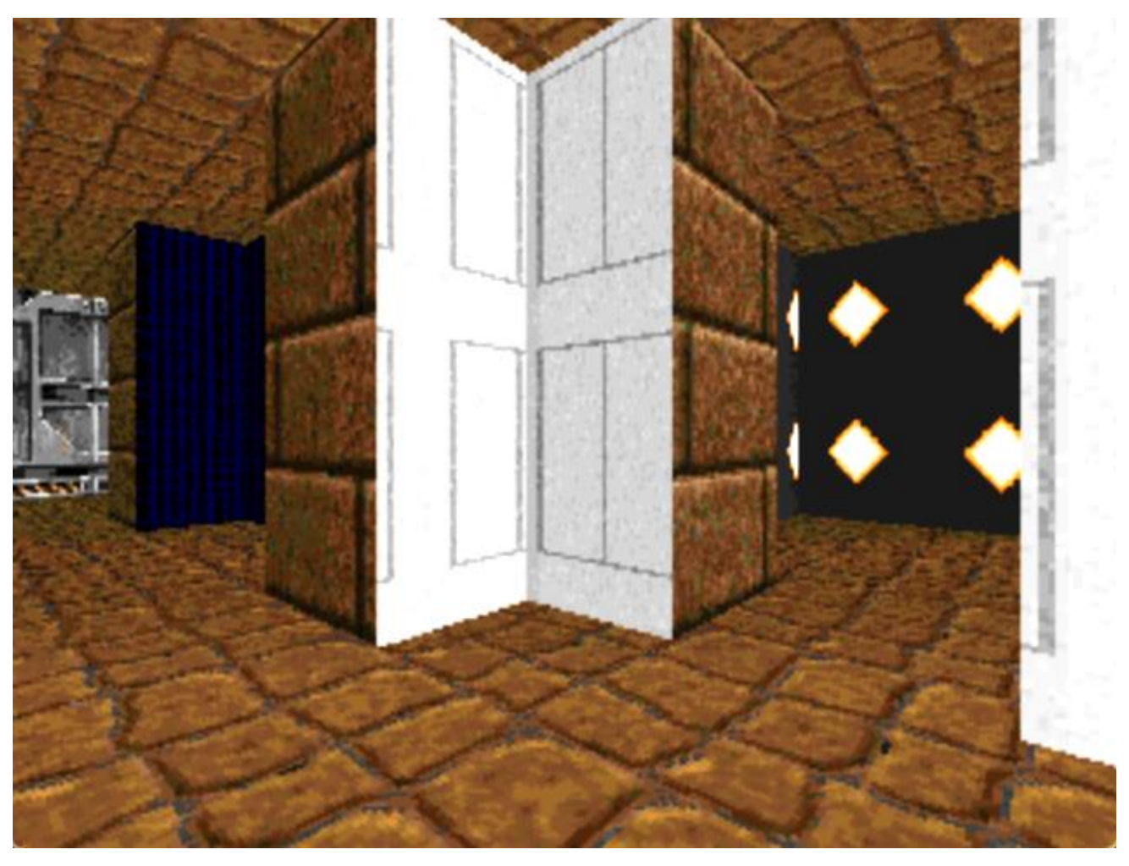
Exploration with Intrinsic Motivation 1/2

Kirill Yankov, Rishan Senanayake, Kilian Heinold, Julien Siems

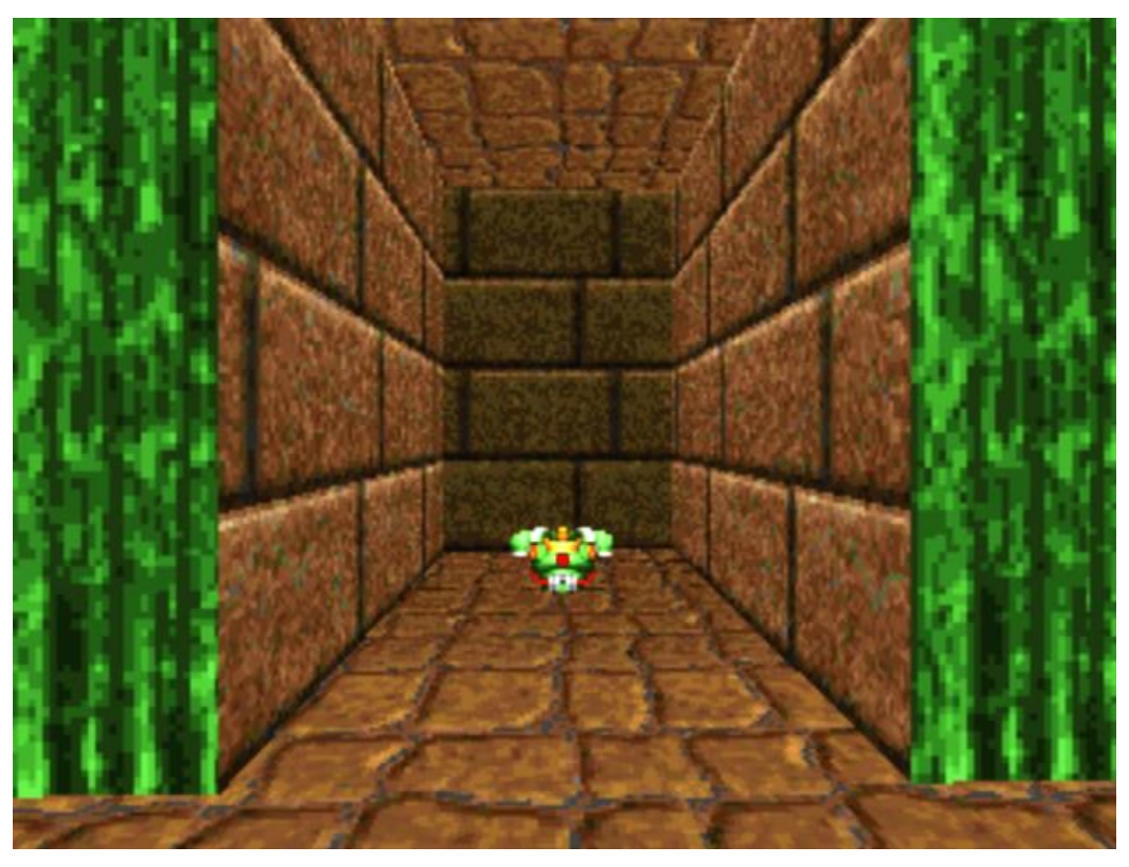
Supervisor: Niklas Wetzel

Motivation

- Evaluate Intrinsic Curiosity Module (ICM) by Pathak et al. [1] on ViZDoom environment [2].
  - What is required to allow a DQN agent to learn in the sparse reward setting?
- Quantify the exploration capabilities of the intrinsic agent
  - How quickly does the agent learn to cover all rooms?
  - How diverse are the trajectories chosen by the agent?



Navigation in the VizDoom environment.



The agent gets the reward only when it reaches the armor at the end of the labyrinth.

Why Intrinsic Motivation?

- Extrinsic Reward is sparse in many real-world applications.
- Sparse extrinsic reward alone not enough to learn some complex tasks.
  - e.g. Montezuma's Revenge on Atari
- Enable agent to learn basic motor skills to navigate an environment.

Curiosity-driven Exploration through Self-supervised Prediction

- **Idea:** If predicting the *next state* given *current action* and *state* is hard then the agent should be more interested in this state.

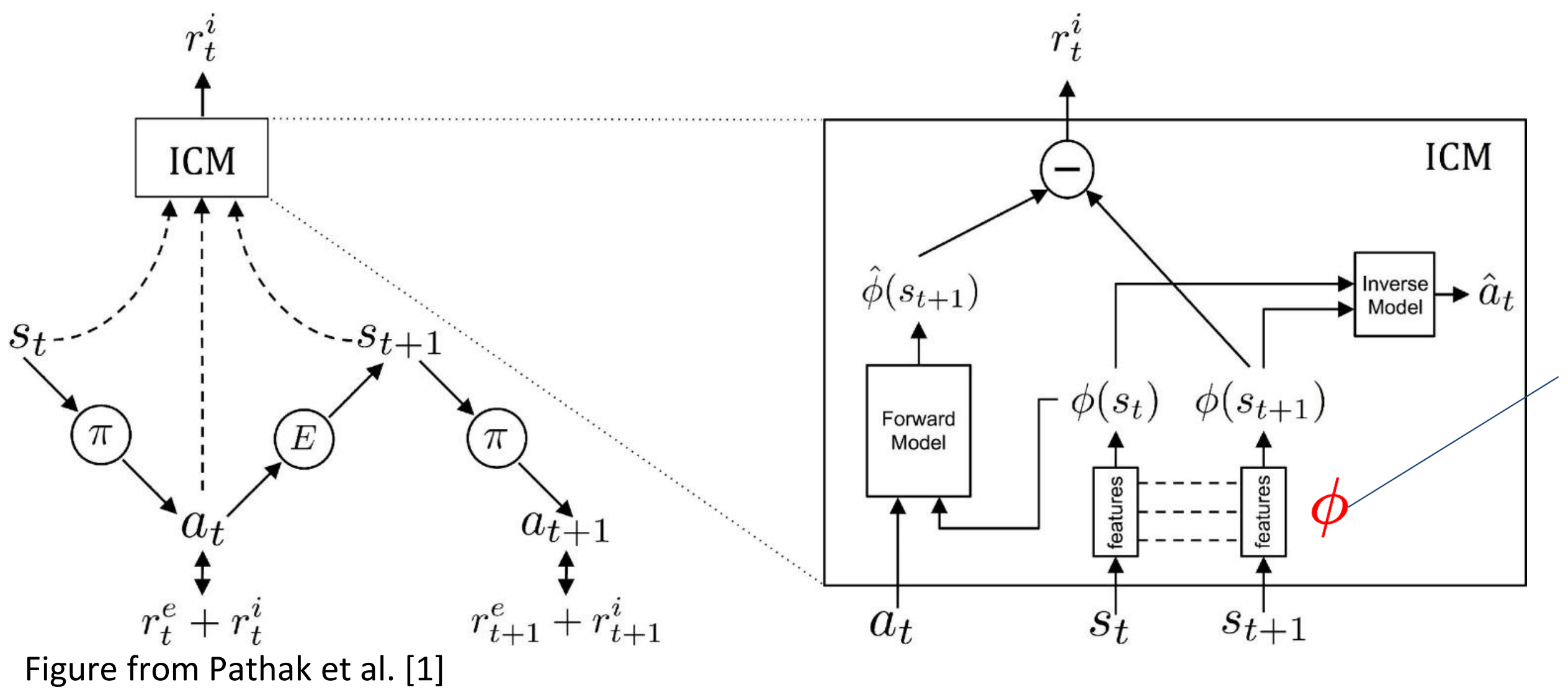


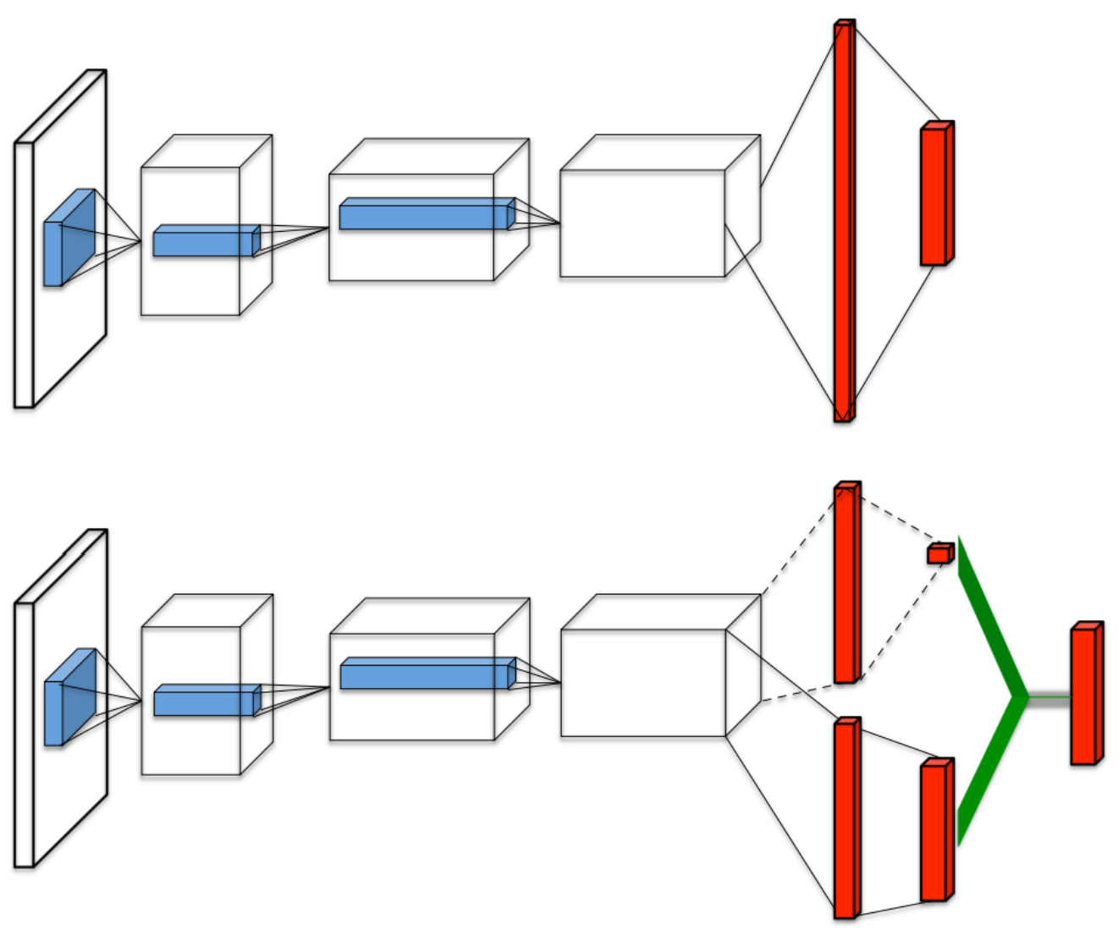
Figure from Pathak et al. [1]

Encoder of frames to lower dimensional feature space (learned)

Inverse Dynamics Model	$\widehat{a_t} = g\left(s_t, s_{t+1}; \theta_I\right)$
Forward Dynamics Model	$\widehat{\phi}(s_{t+1}) = f\left(\phi(s_t), a_t; \theta_F\right)$
Intrinsic Reward Signal	$\boxed{r_t^i = \frac{\eta}{2}\|\|\widehat{\phi}(s_{t+1}) - \phi(s_{t+1})\|\|_2^2}$
Inverse Loss	$L_I = CE(\widehat{a_t}, a_t)$
Forward Loss	$L_F = \frac{1}{2}\|\|\widehat{\phi}(s_{t+1}) - \phi(s_{t+1})\|\|_2^2$

DQN Improvements

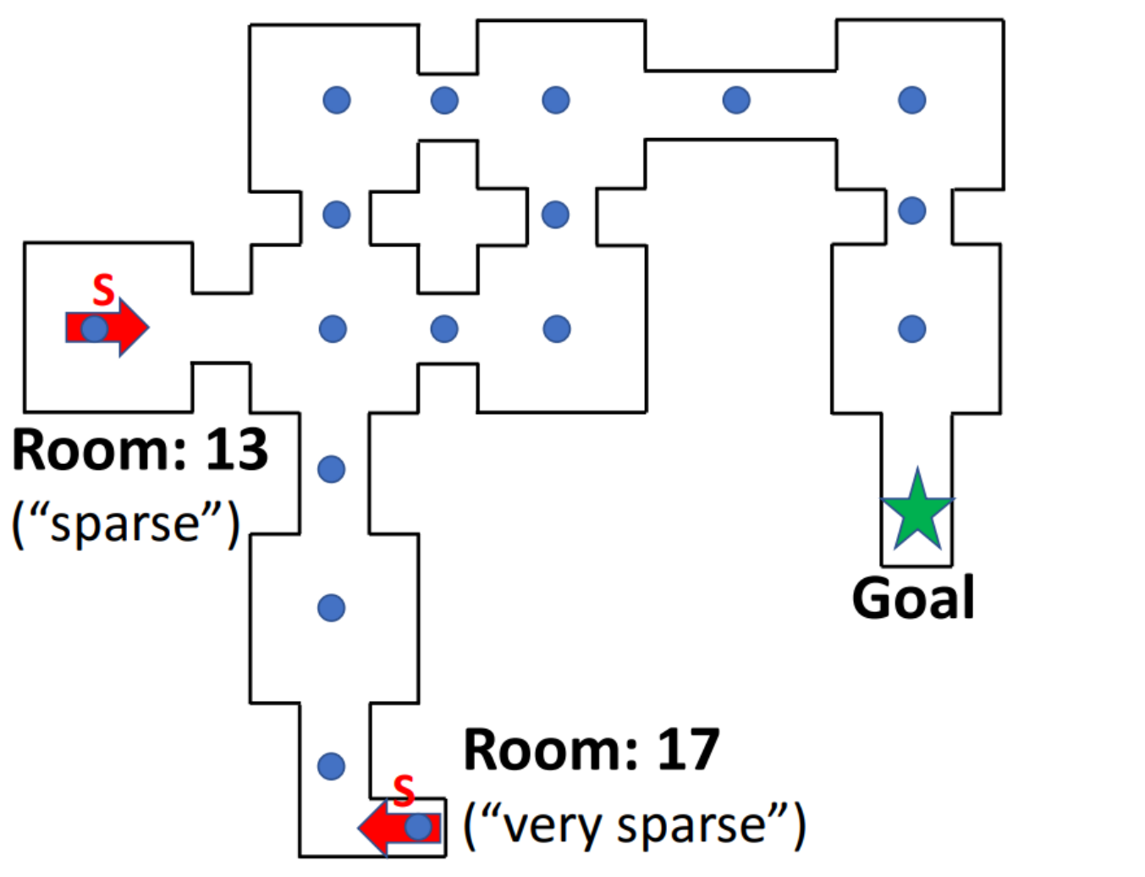
- **Double Q-Learning (Default):** Calculate TD-target with best next action predicted by policy network instead of target network. Counteracts overestimation.
- **Dueling Network:** Network head is replaced by a value and an advantage stream which are summed appropriately in the end.



Wang et al. [4]

- **Multi-Step Reward:** $R_t^{(n)} \equiv \sum_{k=0}^{n-1} \gamma_t^{(k)} R_{t+k+1}$ used for bootstrap targets.
- **Prioritized Experience Replay:** Sample transitions with probability relative to the last encountered absolute TD error.

ViZDoom Environment [2]



Map used in all our experiments. See Pathak et al. [1]

- Based on Doom Engine
- *Dense Map:* Agent spawns randomly in any location marked with blue dot, reward (+1) only at the goal.
- *No Reward Map:* Only spawn in room 17, reward removed entirely

Evaluating Exploration on Training Data

- *Geometric Coverage* - Measures if all the sections have explored sufficiently.

$$H^{\leq n}(m) = \begin{array}{l}\text{Cumulative Visitation Count of room } m \\ \text{until } n^{\text{th}} \text{ Training Step}\end{array}$$

$$\text{Geometric Coverage} = \frac{1}{M}\sum_{m=1}^{M} 1 - \gamma^{H^{\leq n}(m)}$$

- *Distribution Distance* - Measures the distance between cumulative exploration distribution and uniform distribution.

$$U = \text{Uniform Distribution}$$

$$P^{\leq n-1} = \begin{array}{l}\text{Cumulative Visitation Probability Distribution} \\ \text{until } (n-1)^{\text{th}} \text{ Training Step}\end{array}$$

$$\text{Distribution Distance} = Wasserstein(P^{\leq n-1}, U)$$

Evaluating Exploration for Policies

- *Entropy Score* - Measures entropy of weighted Exploration policy.

$$M = \text{Number of Rooms}$$

$$P^{\leq n-1} = \begin{array}{l}\text{Cumulative Visitation Probability Distribution} \\ \text{until } (n-1)^{\text{th}} \text{ Training Step}\end{array}$$

$$\bar{P}^{n} = \begin{array}{l}\text{Estimated Visitation Probability Distribution} \\ \text{of } n^{\text{th}} \text{ Training Step}\end{array}$$

$$P = \alpha\bar{P}^{n} + (1-\alpha)P^{\leq n-1}$$

$$\text{Entropy Score} = -\sum_{m=1}^{M} log(P(m))P(m)$$

- *Exploration Variance* - Measures how targeted the exploration policy is.

$$L = \text{Number of Sampled Trajectories}$$

$$\bar{P}^{n} = \begin{array}{l}\text{Estimated Visitation Probability Distribution} \\ \text{of } n^{\text{th}} \text{ Training Step}\end{array}$$

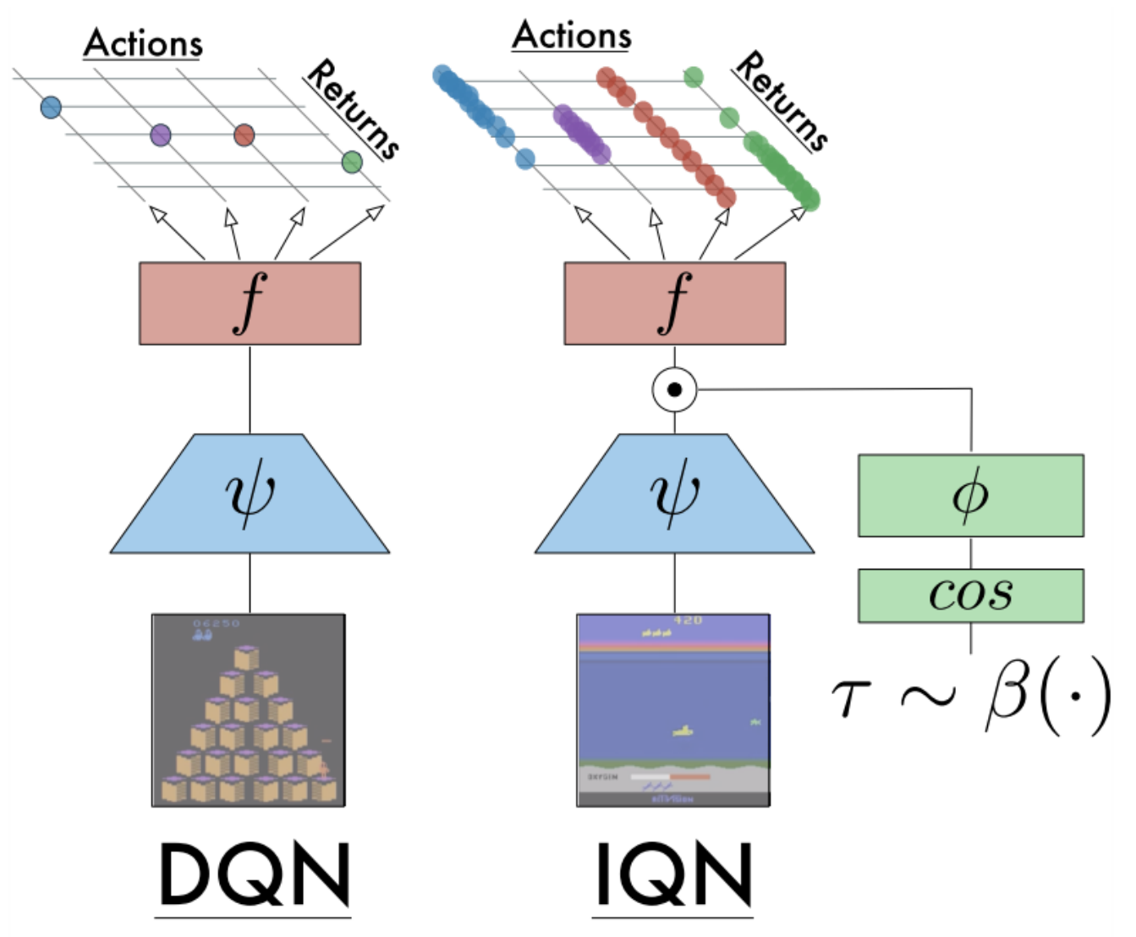
$$T = \text{Number of Time Steps Taken}$$

$$H^{n,(l)} = \begin{array}{l}\text{Visitation Count of } l^{\text{th}} \text{ Sampled Trajectory} \\ \text{in } n^{\text{th}} \text{ Training Step}\end{array}$$

$$\text{Total Variation Distance}(d) = \frac{1}{2}\sum_{m=1}^{M} |P(m) - Q(m)|$$

$$\text{Exploration Variance} = \frac{1}{L-1}\sum_{l=1}^{L} d(\frac{1}{T}H^{n,(l)}, \bar{P}^{n})$$

- **Implicit Quantile Network:** Use quantile regression to approximate the full quantile function for the state-action return distribution.



Dabney et al. [5]

[1] Pathak et al. "Curiosity-driven Exploration by Self-supervised Prediction". In: ICML. 2017.
[2] Kempka et al. "Vizdoom: A doom-based ai research platform for visual reinforcement learning". In: 2016 IEEE Conference on Computational Intelligence and Games (CIG). IEEE. 2016, pp. 1–8
[3] Hessel et al. "Rainbow: Combining improvements in deep reinforcement learning". In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[4] Wang et al. "Dueling network architectures for deep reinforcement learning". In: arXiv preprint arXiv:1511.06581. 2015.
[5] Dabney et al. "Implicit quantile networks for distributional reinforcement learning". In: arXiv preprint arXiv:1806.06923. 2018.

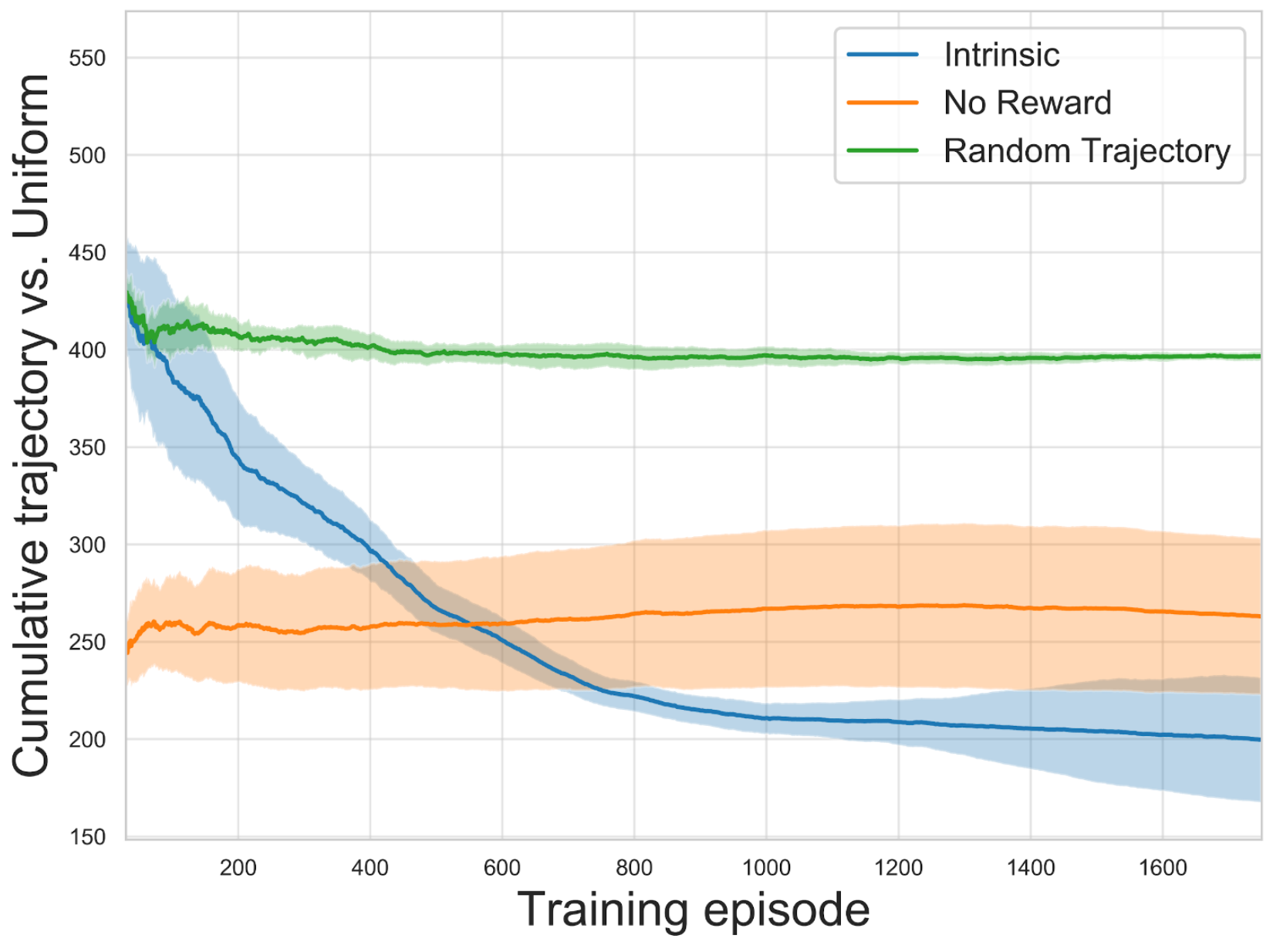
Exploration with Intrinsic Motivation 2/2

Kirill Yankov, Rishan Senanayake, Kilian Heinold, Julien Siems

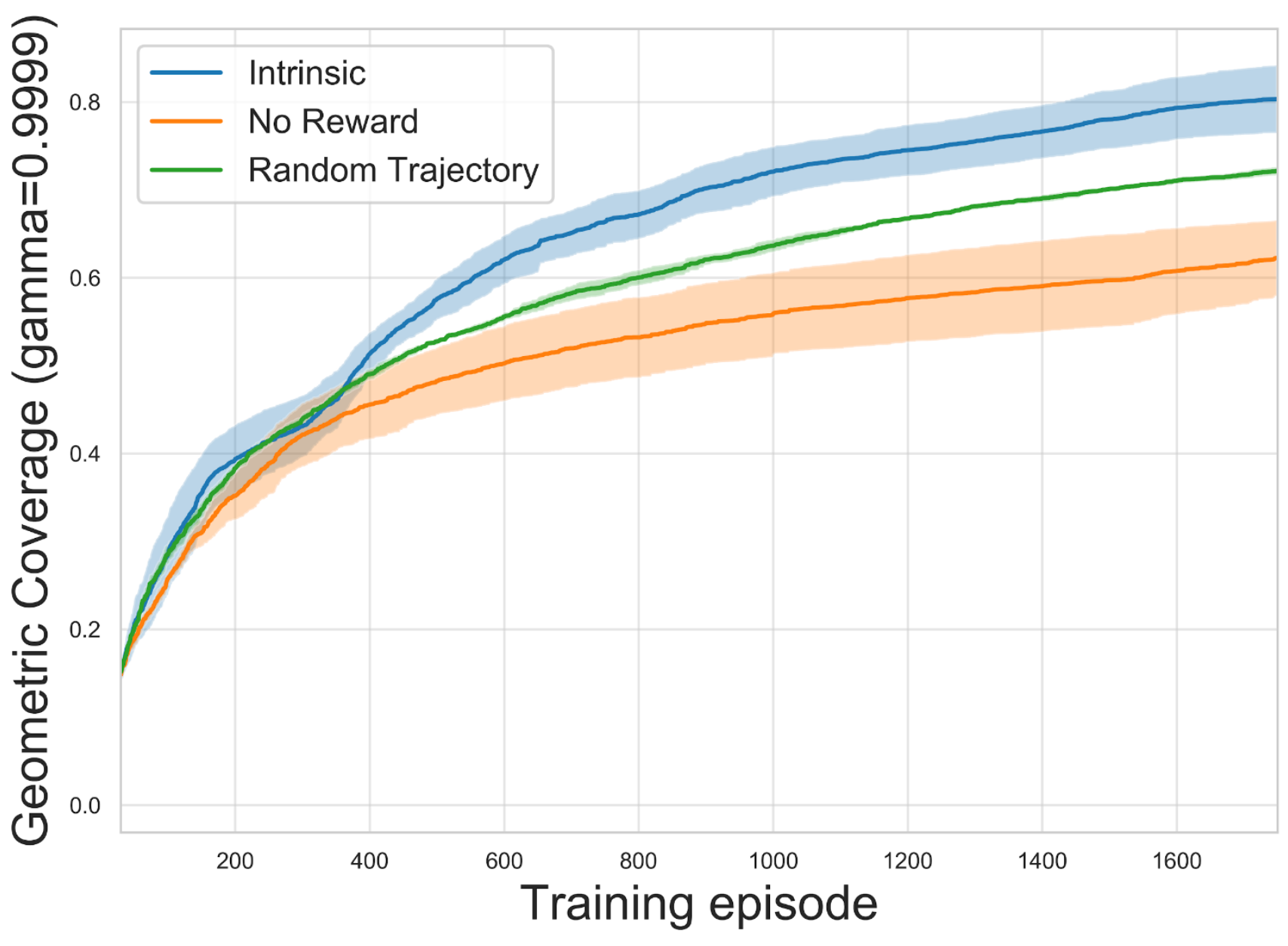
Supervisor: Niklas Wetzel

Experimental Results - No Reward

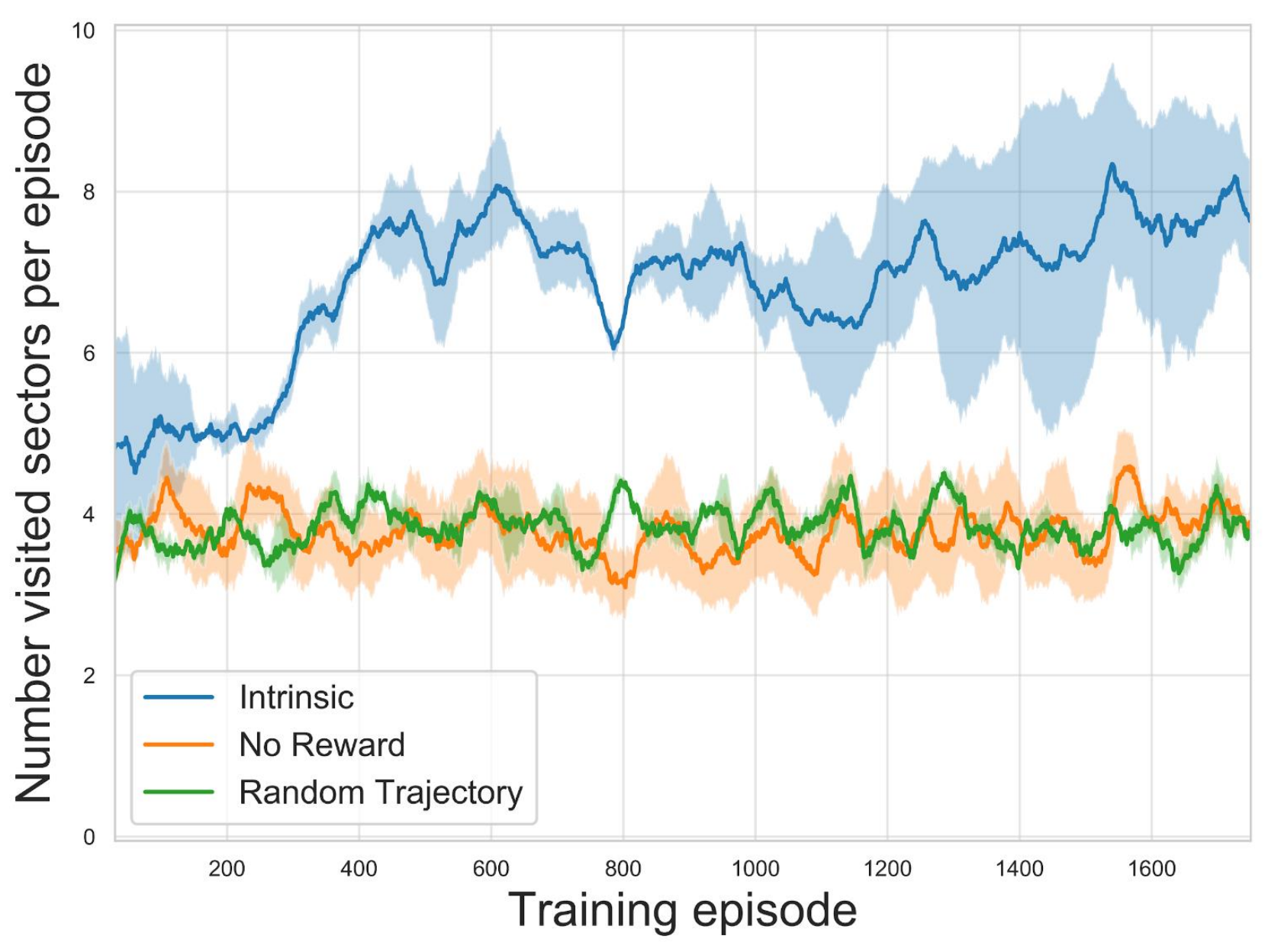
Training:



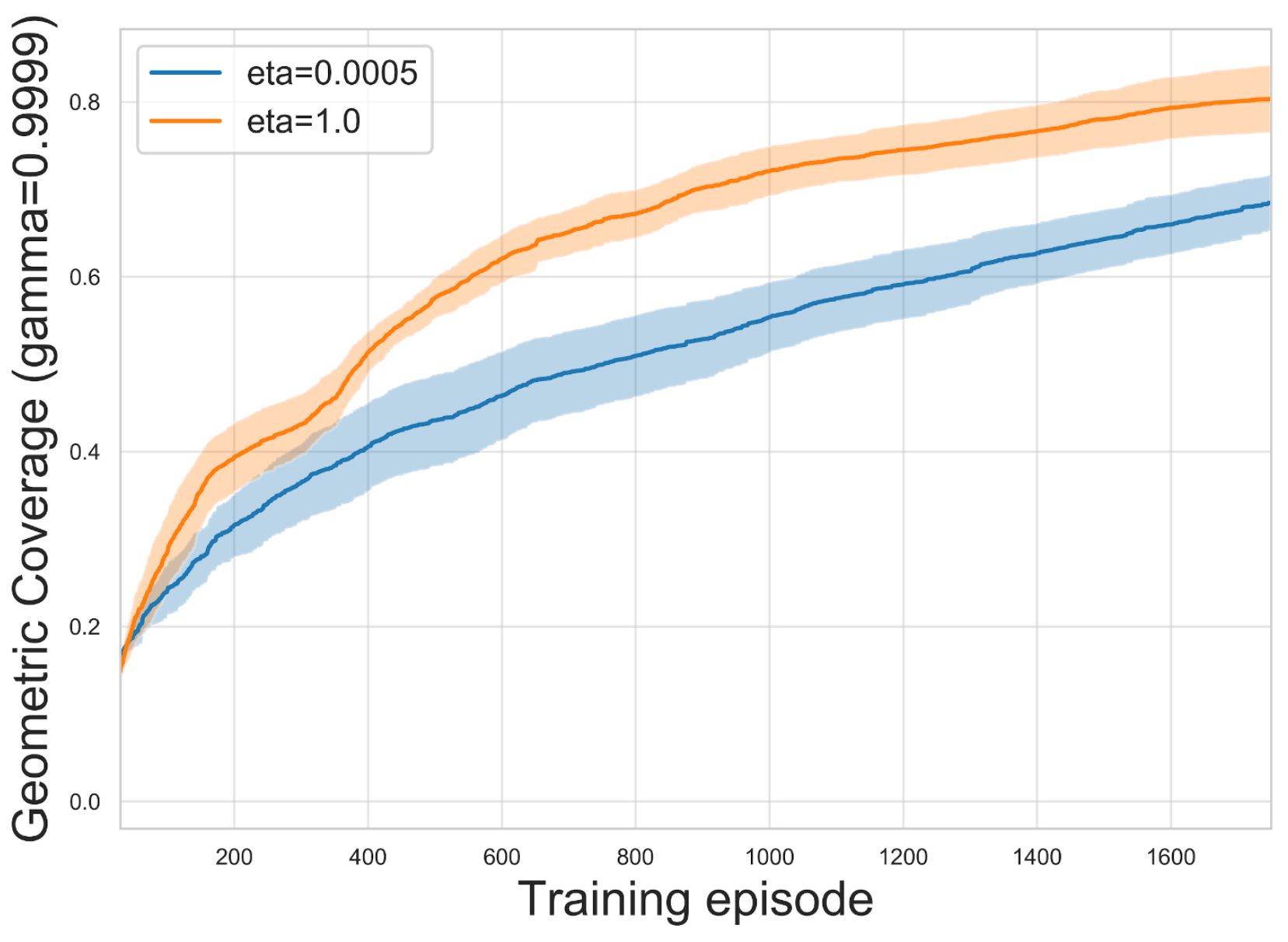
The cumulative trajectory of an agent with intrinsic reward is closest to Uniform Distribution over all rooms.



The agent trained with intrinsic reward covers more rooms quicker than without any reward.

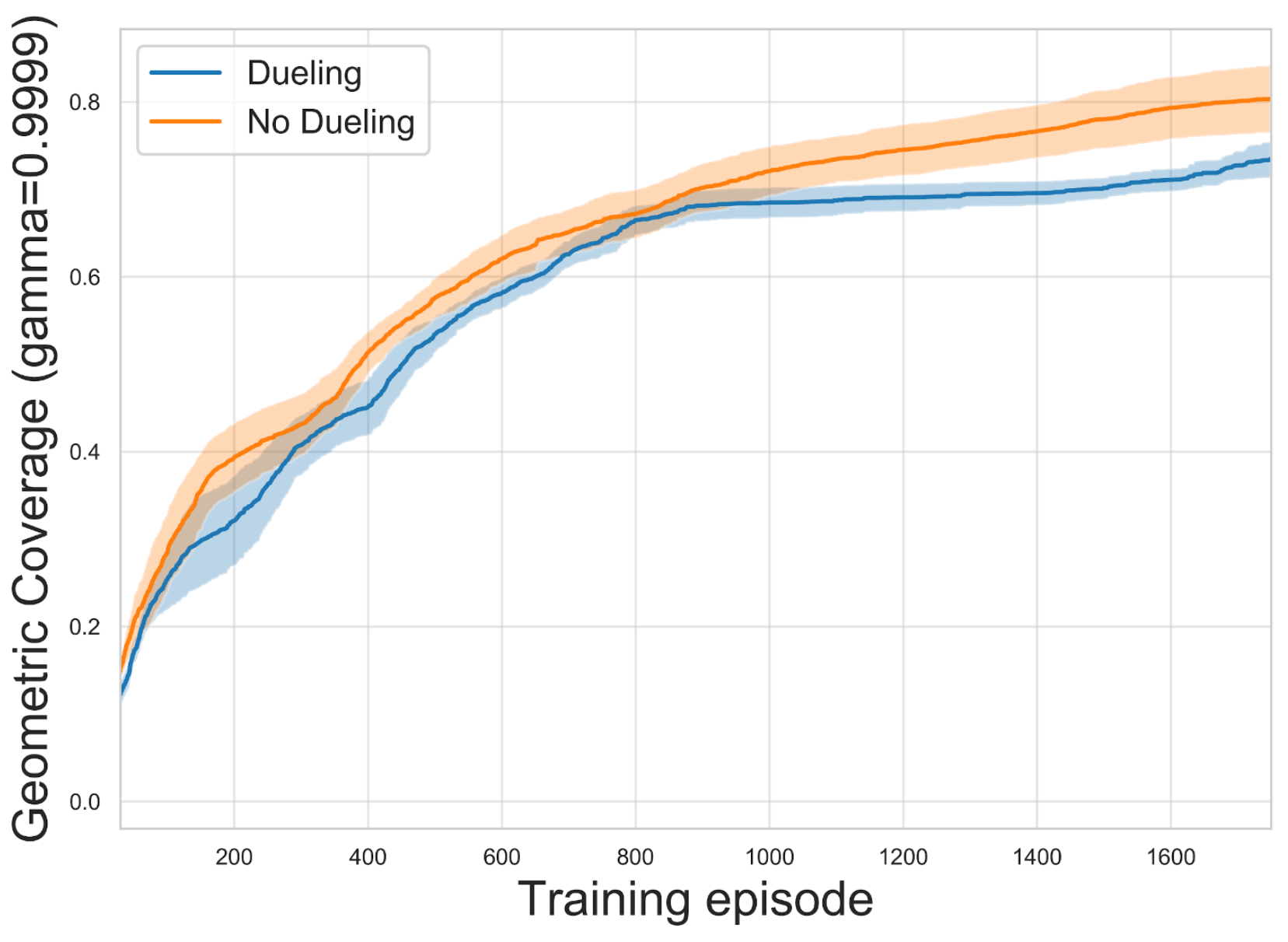


The policy learned with intrinsic reward is capable of reliably exploring more sectors than a random agent or a random trajectory.

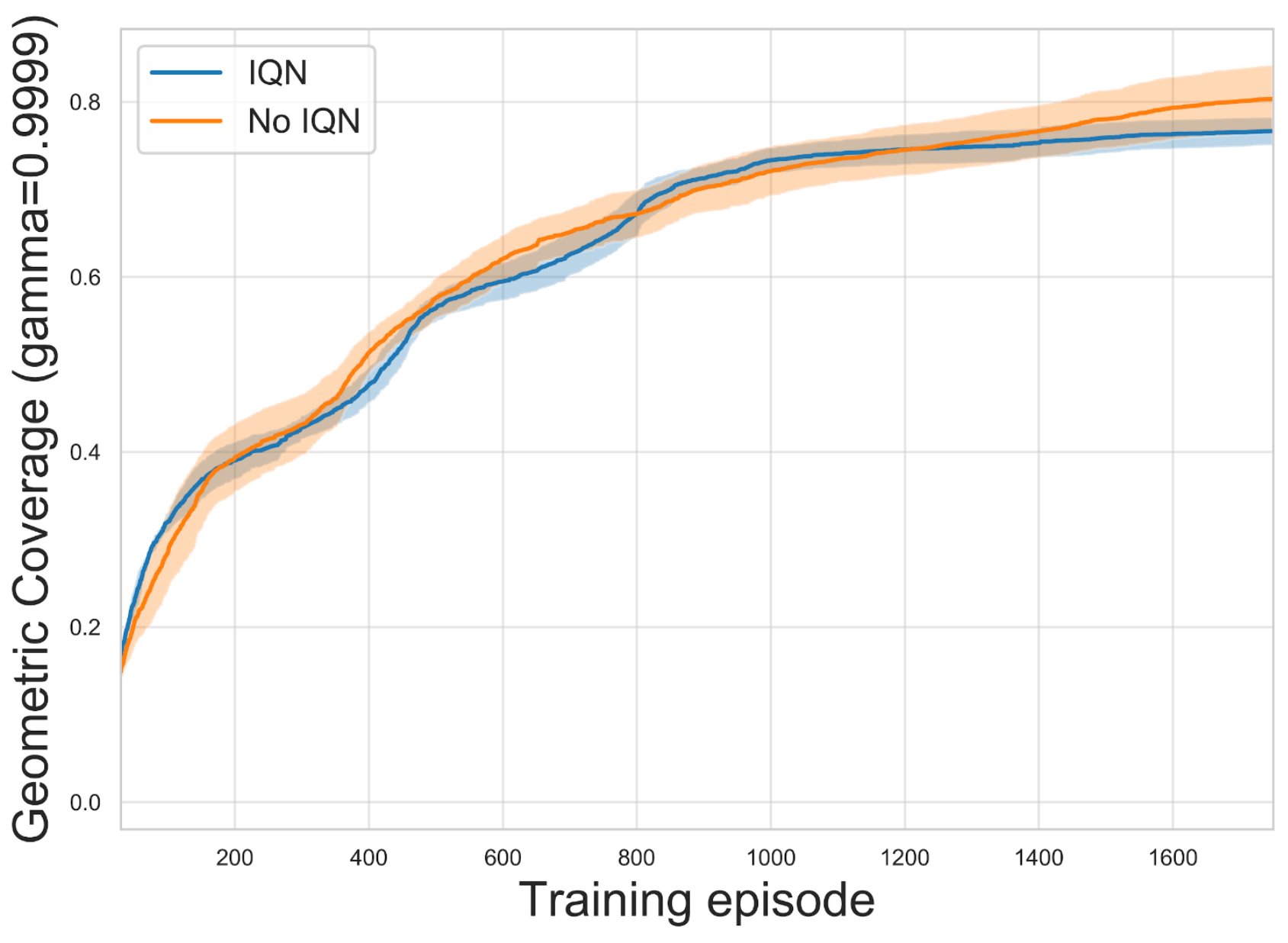


Increased weighting of the intrinsic reward improves the speed of coverage of all rooms. In all high experiments the eta=1.0 was used.

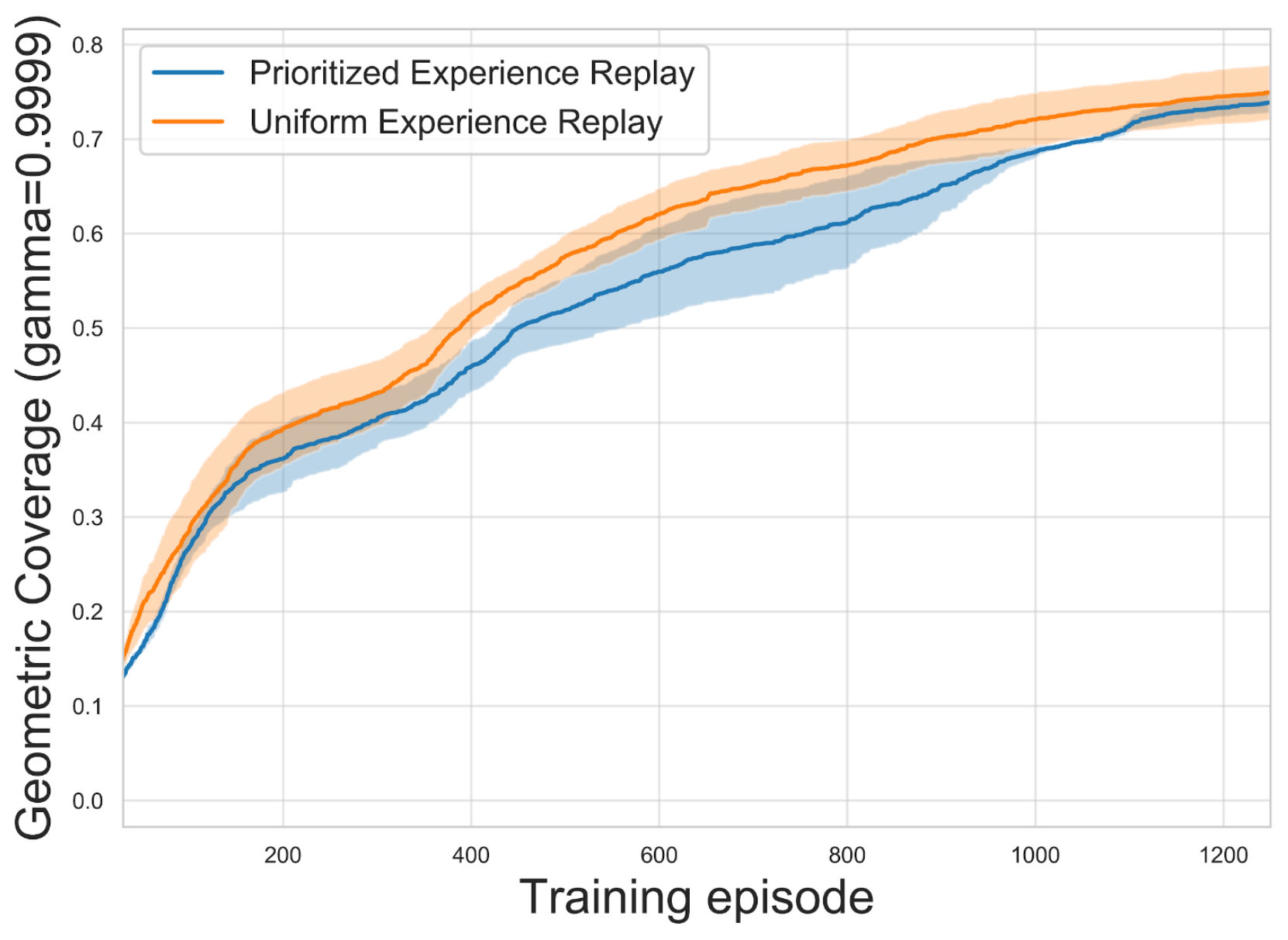
Evaluating DQN improvements:



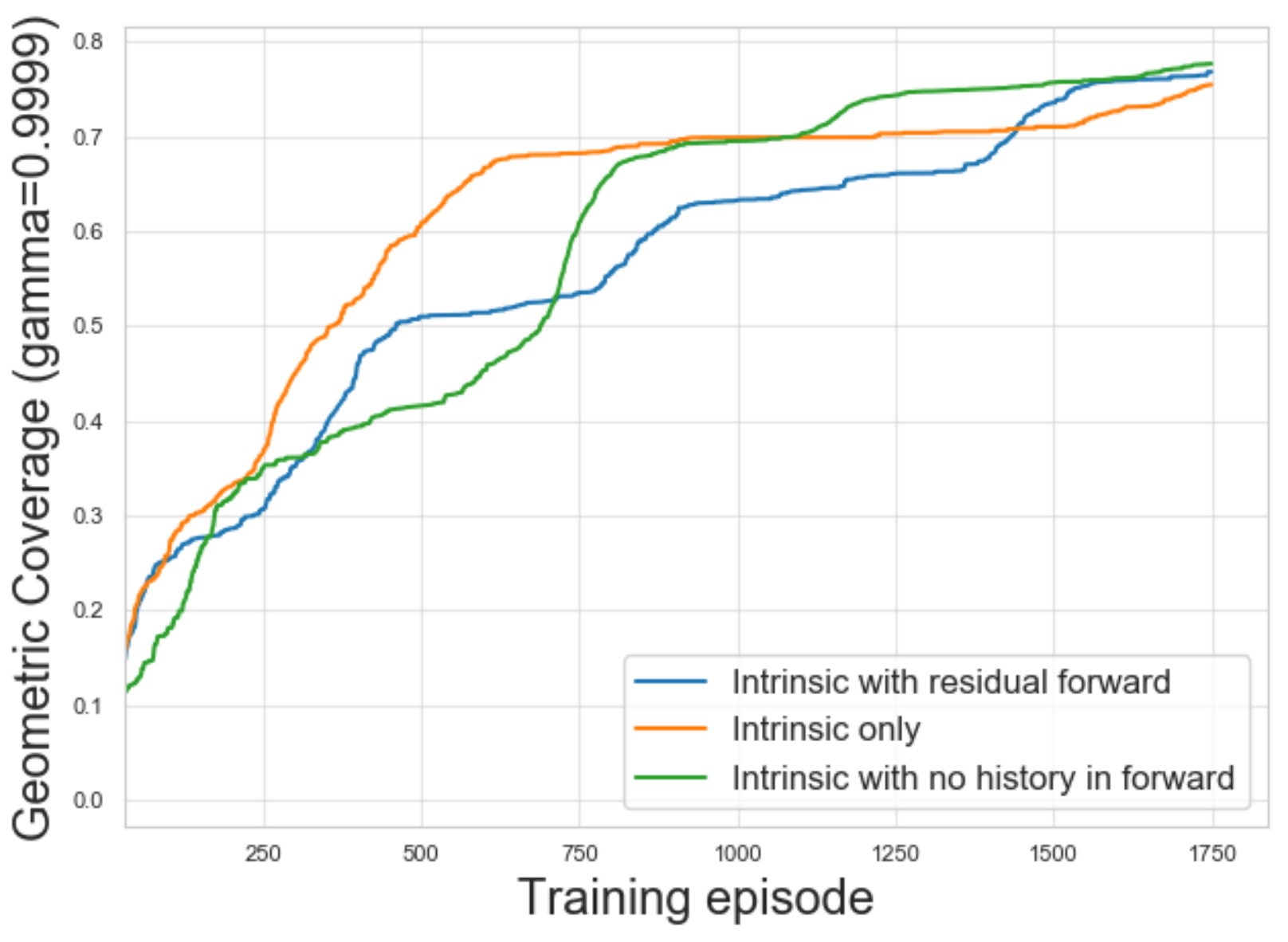
Dueling Networks: Had a minor impact on the training and overall did not improve the exploration capabilities.



Implicit Quantile Networks: Similarly did not improve the overall exploration speed and coverage.



Prioritized Experience Replay: Had no positive impact on exploration coverage and speed.



Residual forward network predicts the difference between current and next embedded states, however they did not improve exploration, the state history was beneficial.

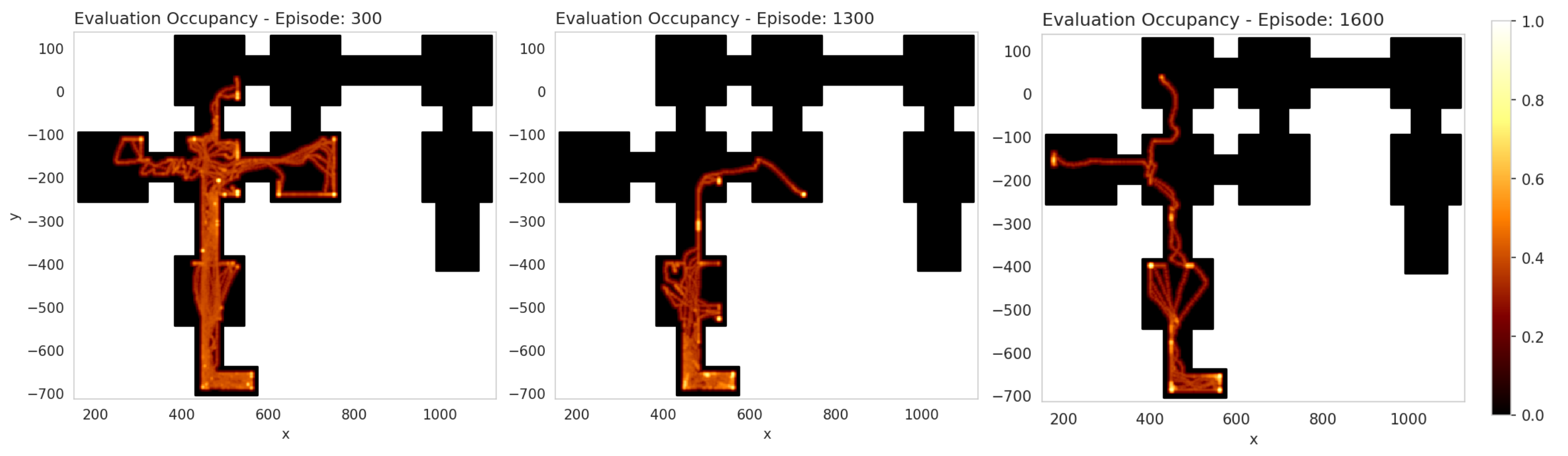
$$f(\phi(s_t), a_t; \theta) = \Delta\hat{\phi} = \phi(s_{t+1}) - \phi(s_t)$$

$$\hat{\phi}(s_{t+1}) = \phi(s_t) + f(\phi(s_t), a_t; \theta)$$

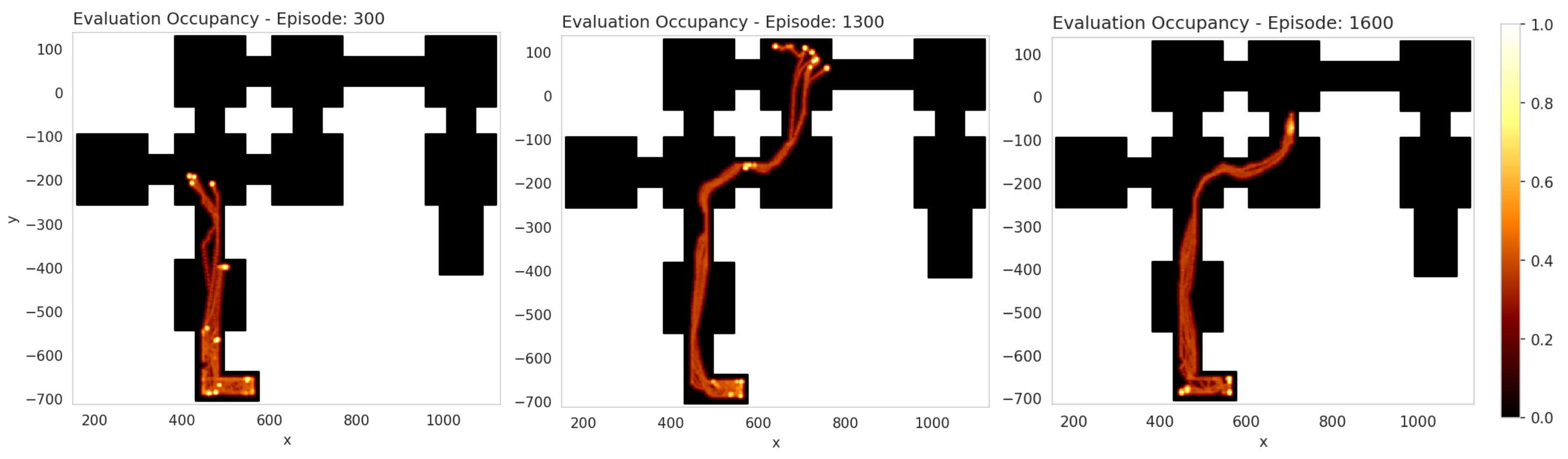
Evaluation:

Evaluation Trajectories: The agent with intrinsic reward follows a more targeted trajectory and avoids walls.

No Reward



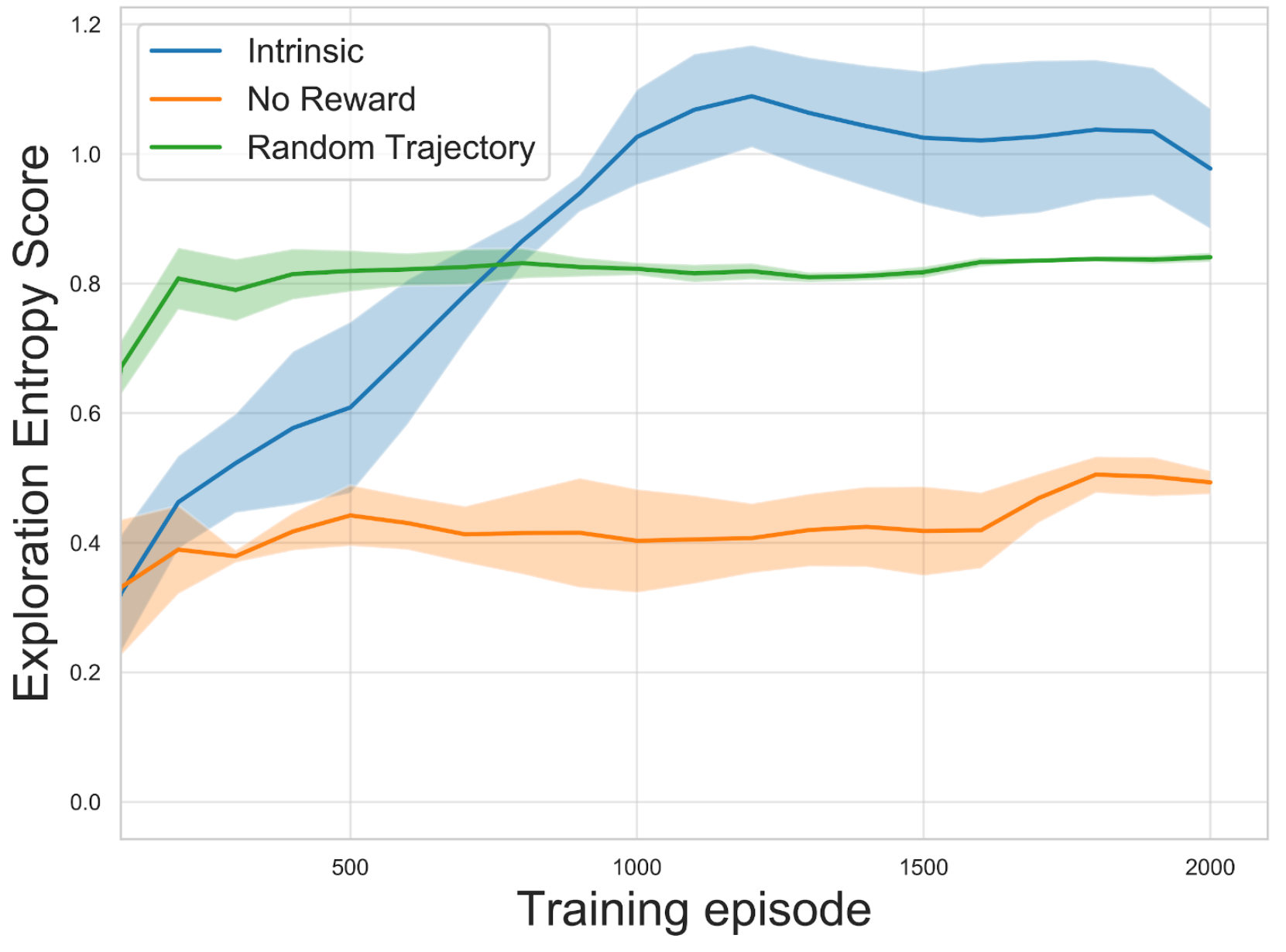
Intrinsic Reward



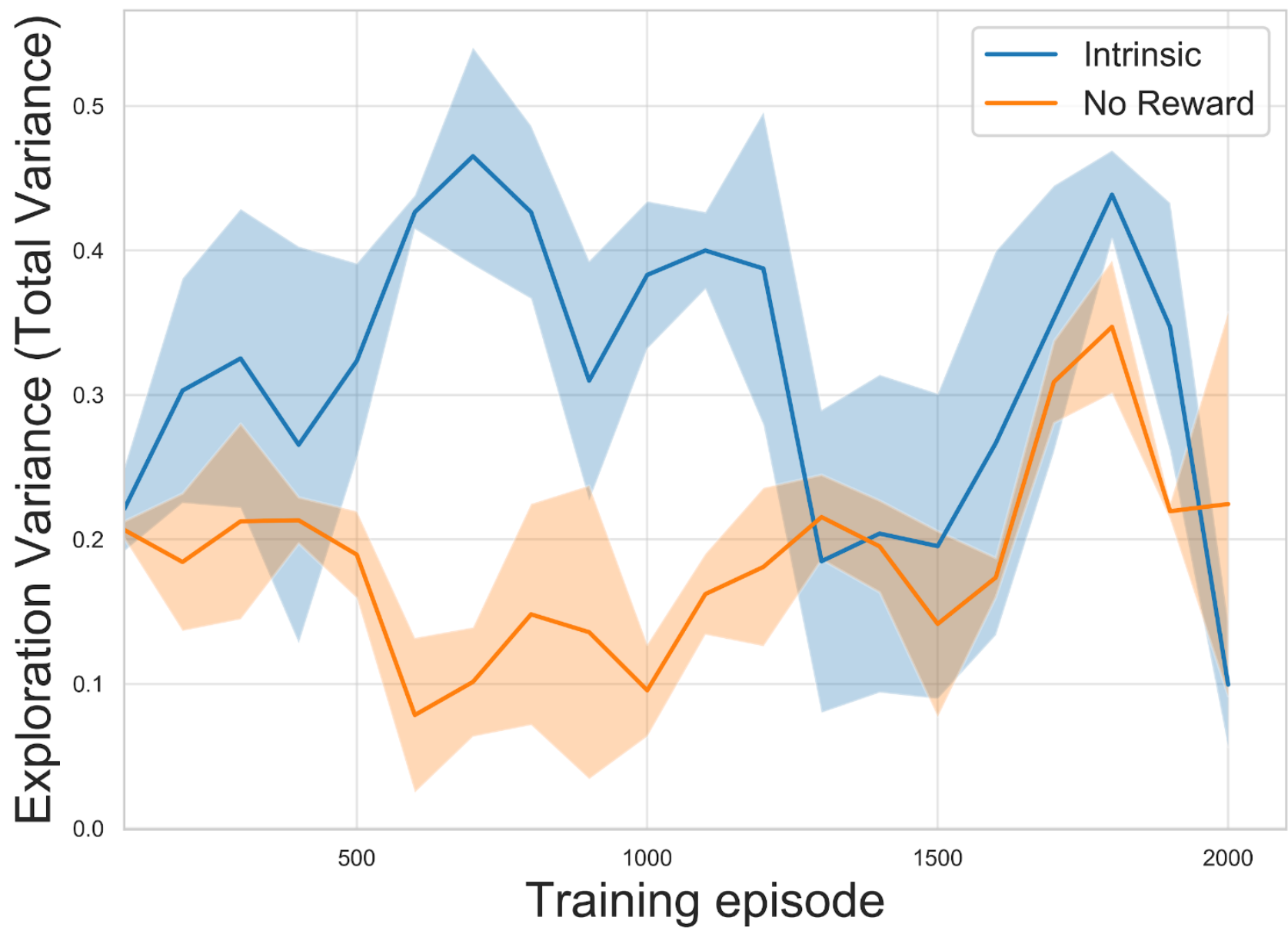
Plotting methodology: Discretized and normalized occupancy values. Averaged over rollouts and slightly blurred. Gamma corrected for better visibility of less occupied regions.

Evaluation metrics:

Averaged over 30 evaluations for every 100 episodes.



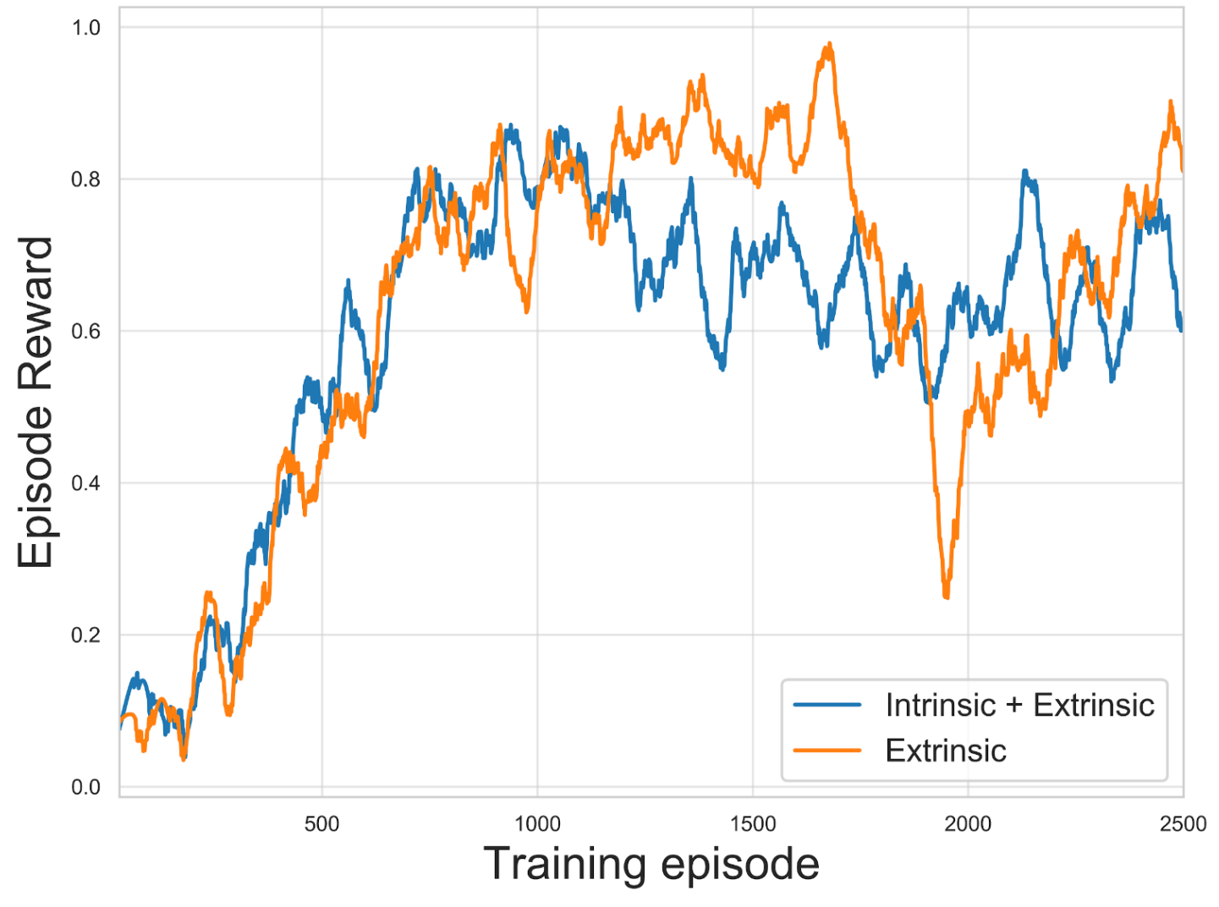
The exploration entropy score during evaluation is higher when using an intrinsic reward.



The exploration variance demonstrates that the policy followed by an agent trained with intrinsic motivation has a less easy to predict outcome.

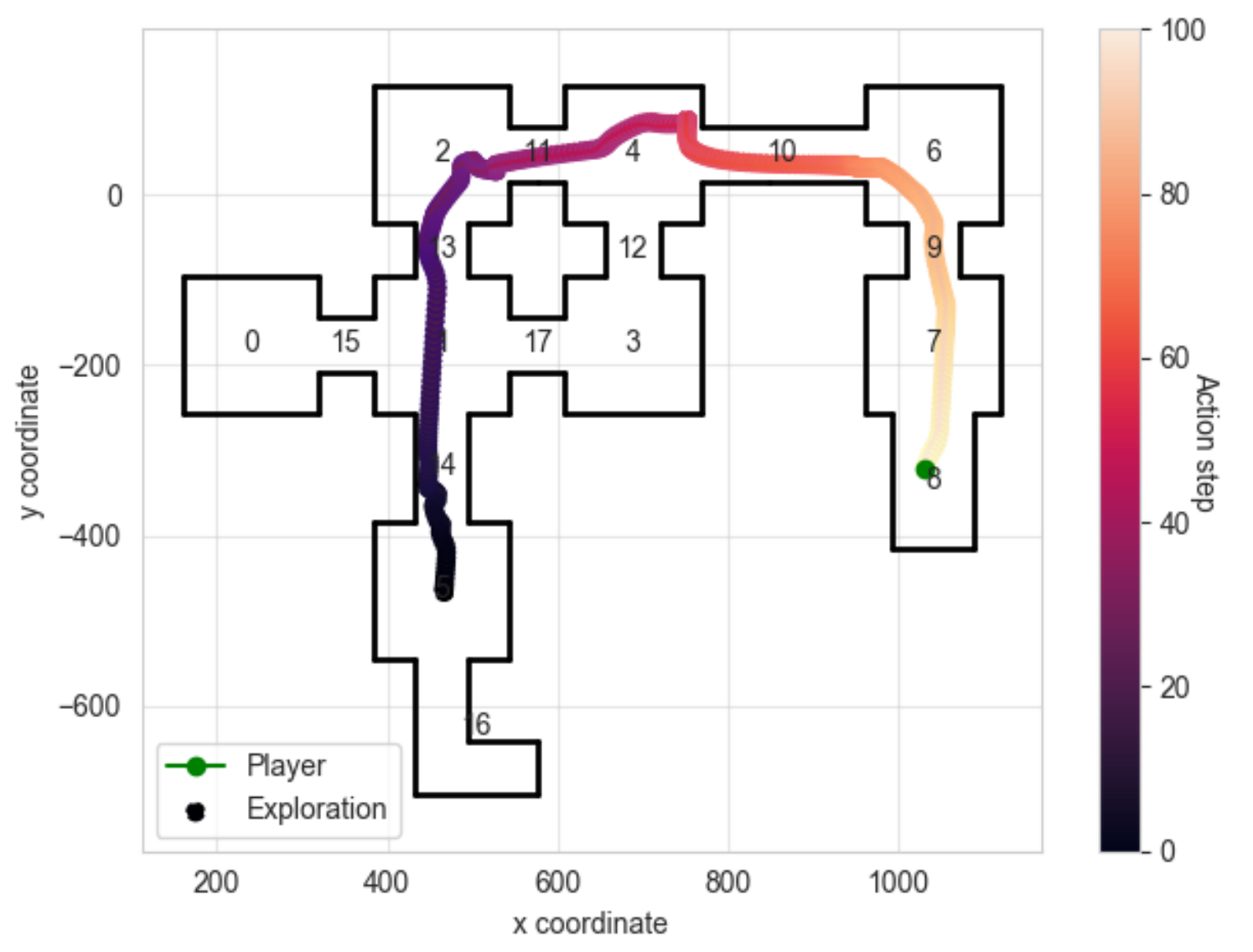
Experimental Results - Dense Reward

- High multi-step reward (50) allowed the reward to be spread along the trajectory to the goal. Low-multi step rewards did not solve the maze.



Intrinsic Reward stabilized the training and alleviated forgetting.

Extrinsic and Extrinsic + Intrinsic Reward learned to solve the maze.



Top-Down-View with trajectory to the goal chosen by the agent.

Intrinsic and extrinsic rewards need to be carefully weighted.

Conclusion

- The average trajectory chosen by the curious agent is closer to the uniform distribution over all sector areas than random trajectories or training without reward.
- Improvements to DQN developed to play Atari games did not help exploration.
- Without any extrinsic reward the agent based on curiosity learns a policy that allows it to cover the sectors very quickly.
- Intrinsic Reward + Extrinsic Reward on the dense reward map learns a stable policy, that suffers less from forgetting, if correctly weighted
- High multi-step reward is necessary in dense reward map to learn a reliable policy