From Task Distributions to Expected Paths Lengths Distributions: Value Function Initialization in Sparse Reward Environments for Lifelong Reinforcement Learning

Soumia Mehimeh * and Xianglong Tang

Posted Date: 6 February 2025

doi: 10.20944/preprints202502.0392.v1

Keywords: reinforcement learning; lifelong learning; statistical reinforcement learning; value function initialization

 doi:10.20944/preprints202502.0392.v1

Article

From Task Distributions to Expected Paths Lengths Distributions: Value Function Initialization in Sparse Reward Environments for Lifelong Reinforcement Learning

Soumia Mehimeh * and Xianglong Tang

School of Computer Science and Technology, Harbin Institute of Technology, 92 West Dazhi Street, Nangang District, Harbin 150001, China

* Correspondence: soumia.mehimeh@gmail.com

Abstract: This paper studies value function transfer within reinforcement learning frameworks, focusing on tasks continuously assigned to an agent through a probabilistic distribution. Specifically, we focus on environments characterized by sparse rewards with a terminal goal. Initially, we propose and theoretically demonstrate that the distribution of the computed value function from such environments, whether in cases where the goals or the dynamics are changing across tasks, can be reformulated as the distribution of the number of steps to the goal generated by their optimal policies, which we name *expected optimal path length*. To test our propositions, we hypothesize that the distribution of the expected optimal path lengths resulting from the task distribution is normal. This claim leads us to propose that if the distribution is normal, then the distribution of the value function follows a log-normal pattern. Leveraging this insight, we introduce "LogQInit" as a novel value function transfer method, based on the properties of log-normality. Finally, we run experiments on a scenario of goals and dynamics distributions, validate our proposition by providing an a dequate analysis of the results, and demonstrate that LogQInit outperforms existing methods of value function initialization, policy transfer, and reward shaping.

Keywords: reinforcement learning; lifelong learning; statistical reinforcement learning; value function initialization

1. Introduction

Lifelong learning in Reinforcement earning (RL) [25] is a transfer learning paradigm that involves continuously assigning to an agent with a various tasks and it should learn while accumulating knowledge to improve its performance when faced with new tasks [12,26,34]. These tasks are drawn from a probabilistic distribution designed to reflect scenarios the agent is likely to encounter in real-world applications [6,11,21,23]. In practice, the distribution of tasks can design different types of changes in the environment settings, mainly the changes of dynamics and the changes of the reward. For example, dynamics changes in application can be seen as a mechanical or an electrical noise in robot sensors which usually modelled a Gaussian distribution [11,23], or wind disturbances affecting a drone, where wind typically follows a Weibull distribution [17]. Tasks can also differ in terms of rewards and goals, such as in a delivery problem where the agent must deliver to customers in different locations that are spread across the city, which are typically distributed according to a normal distribution [6].

One effective strategy for transferring knowledge to new tasks is to initialize the agent's function before it begins learning [1,19,26]. The initialization objective is to provide a knowledge on early stages of learning in order to enhances both the initial, referred to as jumpstart [26,28], and the overall performance of the agent. The main challenge is therefore to estimate the value function of the new task as accurately as possible from the previous tasks data, in order to reduce the number of samples needed

 doi:10.20944/preprints202502.0392.v1

for training new tasks and enabling the agent to reach optimal solutions more quickly. Following this strategy the agent is seen as if it is trying to predict the value function without interacting with the new environment. In machine learning, predictions are often based on the expected values of a dataset, which can be more accurately estimated when the parameters of the dataset distribution are known. Thus, it is essential to extract information from the distribution of RL tasks to accurately estimate the expected value function.

Previous research has explored using probabilistic information from task distributions, typically tackling empirical distributions through simple averaging [19] or through optimistic initialization of the value function based on hyper-parameters within the probably approximately correct Markov Decision Process framework [1,24]. On another hand, probabilistic information has been explored within the experiences data in meta reinforcement learning scale for policy transfer in deep reinforcement learning [3,22,30]. However, to our knowledge, there has been no research on how to specifically induce the connection between the distribution of the value functions from task distribution in regard.

Therefore, our research focuses on initializing the value function in lifelong learning using the parameters of value function distribution. We specifically examine tasks within the framework of a terminal goal state and sparse reward. We claim the state-action space remains similar across tasks while rewards or dynamics change. Our main idea is based on the insight that since the state-action space is the same for each task, the key difference between different task solutions lies within the policy (and hence the value function), which determines the set of actions needed to reach the goal. If the policies differ, then the sequence of states and actions leading to the goal (referred to as a "path") will also differ in length. We use this insight to propose how to measure and deduce the value function distribution. We mainly provide the following contributions:

- We theoretically demonstrate that for tasks sharing the same state-action space, differences in goals or dynamics can be represented by multiple separate distributions of each state-action which is the distribution of the number of steps required to reach the goal called *"Expected Optimal Path Length"*. We then show that the value function's distribution of each state-action is directly connected to its distribution of the expected optimal path lengths.
- We propose a method for initializing the value function in sparse reward setting using a normal distribution of tasks. Our empirical experiments validate this approach by confirming our proposition about the distribution of the expected path lengths. .

The rest of the paper is structured as follows: Section 2 reviews related works, while Section 3 summerizes the relevant preliminaries of RL. Section 4 introduces the definition of the expected optimal path length. In Section 5, we formulate the initialization problem and examine how the value function distribution is connected the distribution of path lengths in both goal distributions and dynamics distributions. We then proceed to present a case study on the normal distribution of path lengths in section 6, which serves as the foundation for our proposed initialization algorithm, LogQInit. Finally, Section 7 presents our experiments and results.

2. Related Works

Knowledge Transfer: Our research is situated within the broad domain of knowledge transfer in reinforcement learning, as outlined in the literature [12,26,34]. Knowledge transfer methods encompass a spectrum of approaches, ranging from policy transfer [7], network parameters, experience data [9], and trajectory samples [27]. Our work focuses on value function initialization, aiming at the objective of enhancing the jump-start performance of the agent [28]. While the field of transfer reinforcement learning has seen limited exploration of value function initialization, existing studies showed strong evidence that adequate initialization improves the learning on the level of single tasks [2,18] and hence is useful in transfer learning. We categorize two main axis of studies of value initialization as knowledge transfer, one studied value function transfer in environments with different state-action space and different dynamics by computing the bisimulation between target and source task reward and transition functions and transferring the value between the states with small distances [8,13,31].

 doi:10.20944/preprints202502.0392.v1

However, these methods require full access to the environment model, which is not always known to the agent. Additionally, these approaches are known for being computationally exhaustive and only applicable to a limited number of tasks. In contrast, our method depends only on task distribution information and therefore eliminates the need for extensive computations, providing significant results that make it suitable for lifelong learning. Another axis of initialization methods studied the use of optimistic initialization theory; [1] proposed an optimistic initialization strategy based on the Probably Approximately Correct framework [24] parameters to estimate the number of steps that should be learned from scratch before starting the transfer. Meanwhile, [19] suggests that the optimistic initialization idea can be improved by trading between the maximum and the mean of the value function from previous tasks using confidence and uncertainty quantities within the agent parameters. However, these works only studied empirical and uniform distributions and did not track the exact relationships between the value function and the task distribution. To tackle this limitation, our work's objective is to analyse the nature of value function distribution according to the task distribution.

Goal based Sparse Reward Tasks: Since our study addresses challenges in a specific type of environments characterized by sparse rewards and a terminal goal it shares similarities with a body of research known as Goal-Conditioned Reinforcement Learning [14,16], particularly in sparse reward settings. In these scenarios, the agent's objective is to reach a specified goal, with rewards provided exclusively upon reaching the goal. This makes the reward signals sparse and difficult to utilize effectively for learning.

Existing research typically tackles such environments through two main approaches. The first approach treats them as single-task problems with dynamic rewards, employing techniques like generating imaginary goals to augment the training data [4] or reshaping the reward function to provide more frequent feedback [20**?**]. The second approach considers each task individually, using methods such as transfer learning or meta-learning to adapt policies across different tasks. For instance, some studies propose reward shaping based on demonstrations to guide policy transfer [10] or design auxiliary rewards to encourage leveraging policies from previous tasks in new contexts [35]. However, these methods predominantly focus on policy-based deep reinforcement learning and rely on experience data, including transitions and rewards. In contrast, our approach focuses on leveraging the value function. We theoretically demonstrate that the value function inherently encapsulates valuable information about the task distribution in sparse reward scenarios. We argue that this information alone is sufficient to transfer useful knowledge across tasks, eliminating the need for computing dense rewards [32].

Tasks distributional information: Furthermore, our research explores the role of distributional information in transfer reinforcement learning. Most studies addressing this issue are within the discipline of meta-reinforcement learning. For example, [22] uses probabilistic distributions of latent task variables to facilitate transfer between tasks by inferring them from experience. Other works tackle the challenge of distribution shifts between offline meta-training and online meta-testing, proposing in-distribution adaptation methods based on the Bayes-Adaptive MDP framework [3,30]. Some researchers have implemented reward shaping by dynamically adjusting the reward structure according to task distributions [33,35] . In contrast, our focus is on transfer learning rather than meta-learning. We study the distributional information within the values of value function instead of distributions of trajectories or rewards, and we do not involve distinct training and testing phases. Although our work is centered on a tabular setting, it has the potential to be adapted for meta-learning in the future.

3. Preliminaries

Markov Decision Process Reinforcement Learning formalizes the interaction between an agent and its environment using Markov Decision Processe (MDP). An MDP $M = \langle S, A, r, g, T, \mathscr{T}, \gamma \rangle$ is defined by a state space S, an action space A, a reward function $r : S \times A \times S \to \mathbb{R}$, and a transition function

 doi:10.20944/preprints202502.0392.v1

$T : S \times A \times S \rightarrow [0,1]$, which specifies the probability of transitioning to the next state given the current state and action. Additionally, a discount factor $\gamma \in [0,1]$ is used to weigh the importance of future rewards. We added two elements to the standard definition of the MDP which are:

- Terminal state $g \in S$: which represents the goal the agent must achieve. This state is terminal which means the agent can not take further actions from there.
- **Most-likely-next-state function**: defined as ($\mathscr{T} : S \times A \rightarrow S$) is a function that map a state-action to the most likely state among all possible next states such as:

$$\mathscr{T}(s,a) = \arg\max_{s'} T(s,a,s')$$

  For example, in a grid-world scenario, taking an action such as moving left might result in a high probability of transitioning to the adjacent left cell, which can be captured by $\mathscr{T}(s,a)$.

Sparse Binary Reward: Environments defined by a terminal goal are often modelled by using a uniform negative reward to all states, except for the goal state, where the agent receives a positive reward [14,16]. This reward model allows the agent to discover the shortest or most efficient path to the goal. In our work, we use a binary reward function as an alternative formulation which serves the same purpose as the negative sparse reward by also driving the agent toward the shortest path. The reward function assigns a binary value $r(s,a,s') = 0$ if $s' \neq g$ and $r(s,a,s') = r_g$ otherwise and we choose r_g [32]. This binary reward structure will be consistent with our theoretical framework without loss of generality, as this reward function can be scaled if a negative reward is preferred.

Episodes and returns: The agent engages with the MDP through a series of iterations referred to as episodes. During each episode, the agent takes sequential steps within the environment, starting from any initial state. At any given step t, the interaction unfolds a path as follows:

$$\tau = s_t, a_t, s_{t+1}, a_{t+1}, ...$$

where $s_{t+1} \sim T(s,a,s_{t+1})$. The return G_t in a state s is defined as the discounted cumulative reward $r_t = r(s_t, a_t, a_{t+1})$:

$$G_t = r_0 + \gamma r_1 + \gamma^2 r_2 + ... = \sum_{t=0}^{\infty} r_t \gamma^t$$

Value Function: The agent's primary objective is to discover a policy function π that maximizes the expected return, as expressed by the state-action function, denoted by Q:

$$Q^{\pi}(s,a) = \mathbb{E}^{\pi}\Big[\sum_{t=0}^{\infty} r_t \gamma^t\Big]$$

The optimal policy is denoted as π^*, is the best policy and its corresponding optimal function is $Q^*(s,a) = argmax_{\pi} Q^{\pi}(s,a)$, where $\pi^*(s) = argmax_a Q^*(s,a)$.

Lifelong Learning: Lifelong learning refers to a specific setting of transfer learning where the agent is assigned tasks continuously from a distribution namely Ω [1,12]. In RL, each task is equivalent to a distinct MDP such as $M_i \sim \Omega$. The main objective is to extract knowledge from a sample of past learned tasks $\{M_0..M_N\}$ to learn new tasks drawn from the same distribution $M_{N+1..\infty} \sim \Omega$.

4. Expected Optimal Path Length

This section introduces the concept of "expected optimal path length" which technically captures the number of steps required for an agent to reach a goal state under a given policy. Here, we formally define this concept, derive its properties, and explore its implications in value function.

 doi:10.20944/preprints202502.0392.v1

4.1. Definition and Representation of EOPL

Let's consider a policy π. When an agent follows this policy starting from any initial state s_0 and reaches the terminal state goal [1], the generated path (sequence of states and actions) $\tau(\pi)$ is the path that generates from this policy represented as:

$$\tau(\pi) = s_0, \pi(s_0), s_1, \pi(s_1), \ldots, s_{H-1}, \pi(s_{H-1}), s_H \equiv g$$

The term H denotes the step of episode termination. The value of H can be a constant in the case of a finite horizon or an integer representing the length of the sequence if the termination condition is reaching the goal state g, which is our case. In this scenario, the length of the sequence is the expected path length under the stochastic dynamics, expressed as:

$$H = \mathbb{E}_{s_t \sim T(s_{t-1}, a_{t-1}, s_t)} \left[\sum_{s_t = s_1}^{s_t = g} 1 \mid \forall s_t \in \tau(\pi) \right]$$

Intuitively, the optimal sequence of the expected path given a policy π consists of each successor state being the most likely one following the action prescribed by π, i.e., $s_{t+1} = \mathscr{T}(s_t, \pi(s_t))$. This can be represented as:

$$\tau^\star(\pi) = s_0, \pi(s_0), \mathscr{T}(s_0, \pi(s_0)), a_1, \ldots, s_{H-1}, a_{H-1}, s_H \equiv g$$

since the next state is deterministic we remove the expectation and the length of $\tau^*(\pi)$ becomes:

$$H = \left[\sum_{\mathscr{T}_t = \mathscr{T}_1}^{\mathscr{T}_t = g} 1 \mid \forall s_t \in \tau^*(\pi) \right] \; ; \mathscr{T}_t = \mathscr{T}(s_{t-1}, \pi(s_{t-1}))$$

From the latter equation we deduce that the quantity H is dependent on the initial state and the policy. So we introduce a function that provides the number of steps given the policy π and the state s. Since we will be addressing the state-action function Q in the rest of the paper, we define this function to take two inputs, the state and action, and denote the number of steps as 1 for transitioning to the corresponding next state by taking action a, followed by the policy π.

Definition 1. *[Expected Optmal Path Length (EOPL)]*$\mathscr{H} : S \times A \to \mathbb{N}_0$ *is a function representing the EOPL given the state-action pair and the policy, with the following properties:*

1. $\mathscr{H}^\pi(s, a) \geqslant 1$ *if* $s \neq g$ *else* $\mathscr{H}^\pi(s, g) = 0$.
2. $\mathscr{H}^\pi(s, a) = 1 + \mathscr{H}^\pi(s', \pi(s'))$, *where* $s' = \mathscr{T}(s, a)$.

The properties can be interpreted as follows:

- The EOPL is 1 if the most likely successor state following action a from state s is the goal g. The agent cannot take any further actions if it is already in the goal state g, as this state is terminal; thus, the optimal length is 0.
- For any state different from the goal state g, the EOPL is strictly greater than 0.

4.2. Value Function in Term of EOPL

We can draw some properties of value function by considering the characteristics of the MDP in question. For instance, in an MDP with sparse positive reward, where the optimal policy should output the shortest path to the goal, the EOPL of π^* becomes the minimum policy length influences the value function. Based on this, the value function expectation can be rewritten as follows:

[1] In the literature, sequences are typically defined to start from a specific time t and end at time H, with the length of the sequence given by $H - t$. To simplify the notation for later theorems and propositions, we choose to refer to the state at time t as 0. This adjustment relaxes the notation while preserving generality

 doi:10.20944/preprints202502.0392.v1

Proposition 1. *Given $\mathscr{H}^*(s,a)$ as the EOPL of the optimal policy π^*, the expected value function can be formulated as:*

$$Q^*(s,a) = \mathbb{E}^*\Big[\sum_{t=0}^{\infty} r_t\gamma^t\Big] = \mathbb{E}^*\Big[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t\gamma^t\Big].$$

This proposition is proven by showing that, in sparse positive reward settings, $\mathscr{H}^*$ is the minimum possible EOPL among all possible policies, and rewards beyond this achieving the goal state give reward equal to zero, so the summation in the value function can be truncated at $\mathscr{H}^*$. The Appendix A.1 contains the detailed proof.

5. Value Function Distribution in Terms of EOPL Distribution

In this section, we delve into the initialization of the value function, outlining both the theoretical foundations and the intuition behind our approach. We begin by introducing the concept of decomposing the task distribution into EOPL distributions and proceed to provide theoretical proofs for both reward and dynamics distribution scenarios by presenting the mathematical propositions that support our method.

5.1. Value Function Initialization by the Distribution Expectation:

The initialization of value function a method of knowledge transfer where agent makes is assigned of its value function before engaging with the environment namely at episode $\emptyset$ [19] where the initialized value function is denoted as $Q^{\emptyset}$. The initialization is analogous to prediction, which means estimating the closest value possible based on the experiences acquired from past tasks. Hence, given the distribution Ω we express the initialization process of the value function of the new task $M_{new} \sim \Omega$ as the expectation from the value functions from the same distribution. Instead of assigning an entity of value function from the data such as $Q^*_{new} := \mathbb{E}_{M\sim\Omega}[Q^{\emptyset}_M]$, we treat each state-action separately and initialize each state action value of the new function:

$$\forall s,a\ Q^{\emptyset}_{new}(s,a) := \mathbb{E}_{M\sim\Omega}[Q^*(s,a)]$$

The most straightforward approach to calculating the expected value is to use the mean from the data of previously seen tasks [19]. While such a method proves effective, particularly for unknown distributions, we posit that the expectation could be more efficiently derived when the distribution is known. Hence, it is a key solution of the initialized function $Q^{\emptyset}(s,a)$.

5.2. Decomposing the Distribution of Similar State-Action Tasks

Assumption 1. *If an agent is solving a set of tasks within the same state-action space, the primary distinction between the solutions for these tasks at any given state is the number of steps the agent takes to reach the goal.*

This assumption implies that, while the state-action spaces remain consistent across tasks, the differences in task solutions arise from variations in the agent's learned policies and the optimal paths it discovers. Once an agent completes the learning process for a given task, it relies solely on the learned policy and no longer requires access to reward signals or the environment's dynamics.

For a set of tasks within the same state-action space, the agent can compute the EOPLs from any starting state to the goal for each task. By aggregating these EOPLs across tasks, the agent effectively builds a dataset that captures how far each state is from the goal under different task dynamics or reward structures. This dataset can then be fit into a distribution with its own parameters, representing the variation in EOPLs across tasks. We refer the reader to Figure 1, which provides an illustration of the assumption under discussion.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 February 2025 doi:10.20944/preprints202502.0392.v1

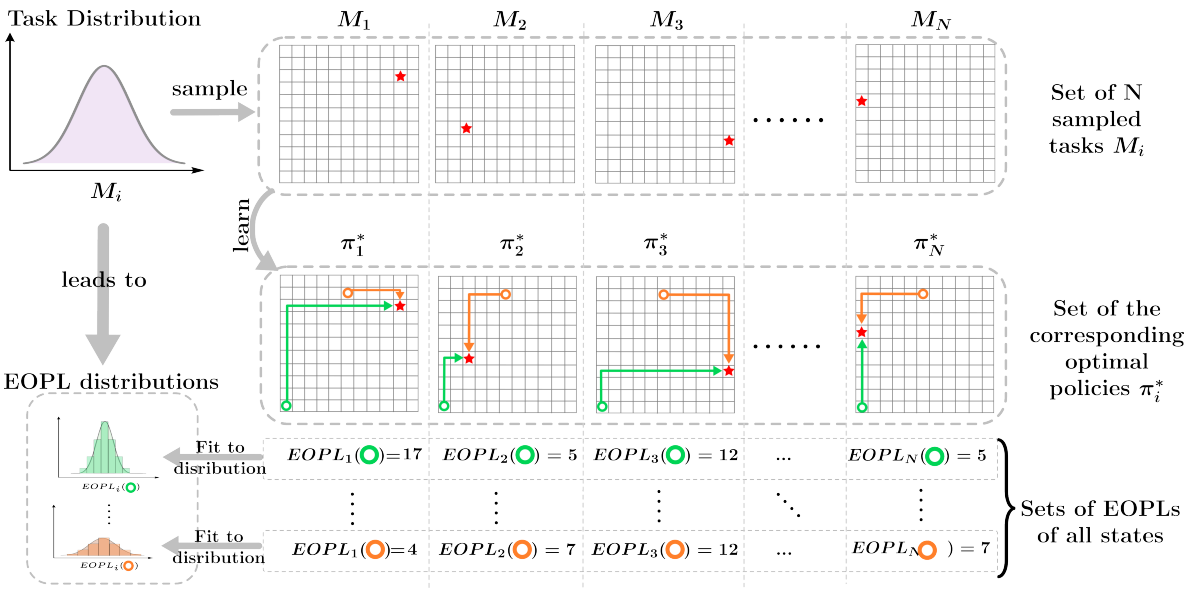


Figure 1. Illustration of the insight of how task distributions can be decomposed into multiple distributions of expected optimal path lengths for all states.

Proposition 2. *Let Ω represent a distribution of tasks $\{M_i \mid M_i \sim \Omega\}$, all sharing the same state-action space. Then, the expected value of the optimal value function over this task distribution can be expressed as follows:*

$$\mathbb{E}_{M\sim\Omega}[Q^*] \equiv \forall s, a \in \mathcal{S}\times\mathcal{A}, \quad \mathbb{E}_{M\sim\Omega}[Q^*(s,a)] = \mathbb{E}_{\mathscr{H}^*(s,a)\sim\mathcal{H}(s,a)}\left[\sum_{t=0}^{\mathscr{H}_i(s,a)} r_t\gamma^t\right],$$

where $\mathcal{H}(s,a)$ represents the distribution of EOPL in (s,a).

To validate this proposition, we analyze independently the case of goals distribution and dynamics distribution. In the following sections, we provide detailed derivations for these cases, demonstrating how the task distribution is composed of all the state-actions EOPL distributions.

5.3. Goal Distribution

The scenario of goal distributions is often related to tasks with the objective of path planning and finding the shortest path to achieve a dynamically changing goal [4,14]. The variation of the goals from a task to another implies the changes of reward. Hence a task sampled from a distribution of goals $M_i \sim \Omega_g$ is written as $M_i = \langle S, A, r_i, g_i, T, \mathscr{T}, \gamma\rangle$.

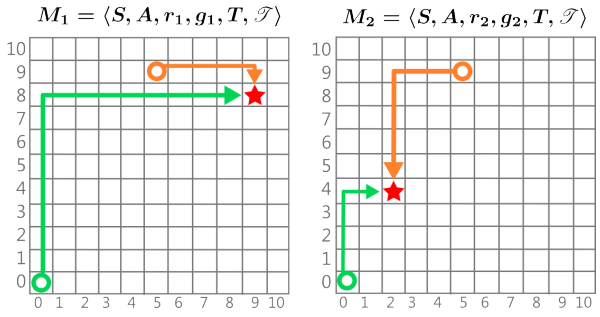


Figure 2. Example of two MDPs with different rewards. A circle represents a state and the arrows represent the EOPL, the star represents the goal. In M_1 with the goal position $g = [8,9]$, we observe $EOPL(\bigcirc) = 17, EOPL(\bigcirc) = 4$. In M_2 the goal position is $g = [4,2]$ therefore $EOPL(\bigcirc) = 5, EOPL(\bigcirc) = 7$.

 doi:10.20944/preprints202502.0392.v1

Proposition 3. *if $M \sim \Omega_g$, then $\forall s, a$ there exists a distribution $\mathcal{H}(s,a)$ such as:*

$$\mathbb{E}_{M_i \sim \Omega_g}[Q^*(s,a)] = \mathbb{E}_{\mathscr{H}_i^*(s,a) \sim \mathcal{H}(s,a)} \mathbb{E}^* \left[\sum_{t=0}^{\mathscr{H}_i^*(s,a)} r_t \gamma^t \right]$$

Proof. Suppose we have a distribution Ω_g where $\forall M_i \in \Omega_g \;\; M_i = \langle S, A, r_i, g_i, T, \mathscr{T} \rangle$. Intuitively, the same state on the different tasks will be at different distances away from the goal and therefore if we have

Lemma 1. *M_i and M_j two tasks from the distribution Ω where $g_i \neq g_j$ then there exists at least one s, a where $\mathscr{H}_i^*(s,a) \neq \mathscr{H}_j^*(s,a)$*

Proof. According to the properties of $\mathscr{H}$ we have $\mathscr{H}_i^*(g_i, a) = 0$ since g_i is terminal and the agent can't take any further action , on the other hand, we $\mathscr{H}_j^*(g_i, a) > 0$ since $g_i \neq g_j$ therefore g_i is not terminal in M_j there distance between the state g_i in M_j and its goal g_j is non-zero, i.e.,. Consequently, we have $\mathscr{H}_i^*(g_i, a) \neq \mathscr{H}_j^*(g_i, a)$, establishing the validity of the proposition. □

Therefore , we can extract the set of $\mathscr{H}_i^*$:

$$\mathcal{H}(s,a) = \left\{ \mathscr{H}_i^*(s,a), \forall M_i \in \Omega_g \right\}$$

For any new task from the distribution, the expectation of its value function can be reformulated in terms of its horizon distribution:

$$\mathbb{E}_{M \sim \Omega_g}[Q^*(s,a)] = \mathbb{E}_{M \sim \Omega_g} \mathbb{E}^* \left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t \gamma^t \right] = \mathbb{E}_{\mathscr{H}^*(s,a) \sim \mathcal{H}(s,a)} \mathbb{E}^* \left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t \gamma^t \right]$$

This establishes that the expectation of the value function $Q^*(s,a)$ is dependent on EOPL distribution $\mathscr{H}^*$ sampled from the tasks distribution Ω_g. □

5.4. Dynamics Distribution

Environment dynamics refer to the rules that govern how the agent moves to the next state in response to the actions taken in the environment. Formally, in an MDP, the dynamics function T determines the probabilities of transitioning to all states in the environment after taking action a from any state s. This function assigns probabilities mainly to neighbouring states that can be directly accessed by available actions, with all other states having probabilities of 0 since they are inaccessible from that state. Changes in dynamics within tasks sharing the same state and action space can be categorized into two scenarios:

1. **Different T Similar $\mathscr{T}$:** Only the probability of transitioning to the same neighbouring state using the same action changes slightly. In this case, $\mathscr{T}(s,a)$ remains nearly the same, so the EOPL does not significantly change across tasks.
2. **Different T Different $\mathscr{T}$:** The same state-action pair may result in transitions to different states in different tasks. In this case, the EOPL changes depending on whether the transition moves the agent closer to the target or farther away. For example, consider a drone in a windy environment [17], where the target is always the same. Due to varying wind conditions, the drone will take different paths to reach the target each time. If the wind pushes it towards the target, it will reach the target more quickly; if the wind pushes against it, the drone will take longer to achieve the goal.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 February 2025 doi:10.20944/preprints202502.0392.v1

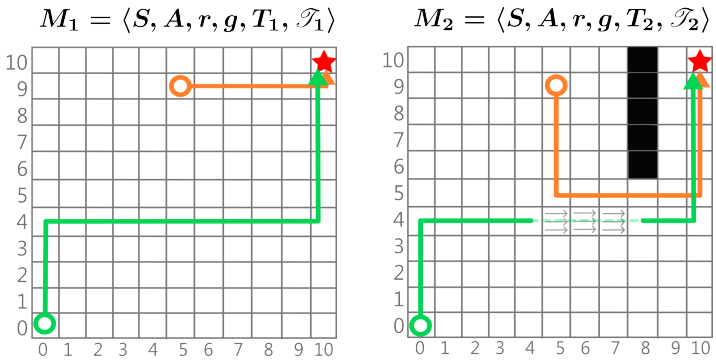


Figure 3. Example illustrating two MDPs with different dynamics. A circle represents a state, the arrows are the EOPL, and the star represents the goal. Here, we observe $EOPL(\bigcirc) = 19$ and $EOPL(\bigcirc) = 6$. In M_2, new dynamics are introduced, such as the transition from state $[6, 8]$ to $[10, 8]$, representing an impassable wall, which results in a longer EOPL of $EOPL(\bigcirc) = 15$ compared to M_1. Additionally, in the transition from $[4, 5]$ to $[4, 7]$, a pushing force moves the agent three states to the right, reducing $EOPL(\bigcirc) = 16$ and making it shorter than in M_1.

Let Ω_T be a distribution of tasks with different transitions such as $\forall M_i \in \Omega_T$, $M_i = \langle S, A, r, g, T_i, \mathscr{T}_i, \rangle$. We consider g to be unchanged over the tasks but without loss of generality because this assumption helps prove mathematically that the change of dynamics is a change that can be translated as a change in EOPL.

Proposition 4. *If $M \sim \Omega_T$, $\forall s, a$ there exists a distribution $\mathcal{H}(s, a)$ such that:*

$$\mathbb{E}_{M \sim \Omega_T}[Q^*(s, a)] = \mathbb{E}_{\mathscr{H}(s,a) \sim \mathcal{H}(s,a)} \mathbb{E}^* \left[\sum_{t=0}^{\mathscr{H}(s,a)} r_t \gamma^t \right]$$

Proof. Unlike the proposition concerning goal distribution, we cannot assert with certainty that each $\mathscr{H}_i$ is unique, as this depends on the degree of variance in dynamics across tasks and their similarities. Therefore, we limit our claim to demonstrating that each task certainly possesses a unique $\mathscr{H}_i$ due to the lack of conclusive evidence to the contrary, as outlined in the following lemma.

Lemma 2. *For M_i and M_j as two tasks from the distribution Ω_T where $g_i = g_j$ and $\mathscr{T}_i \neq \mathscr{T}_j$, each $M_i \in \Omega_T$ has its unique $\mathscr{H}$.*

Lemma 2. Since $\mathscr{T}_i \neq \mathscr{T}_j$, there exists at least one s, a such that $\mathscr{T}_i(s, a) \neq \mathscr{T}_j(s, a)$. Therefore, using Property 4 of the definition:

$$\mathscr{H}_i(s, a) = 1 + \mathscr{H}_i(\mathscr{T}_i(s, a), \pi_i(\mathscr{T}_i(s, a)))$$

$$\mathscr{H}_j(s, a) = 1 + \mathscr{H}_j(\mathscr{T}_j(s, a), \pi(\mathscr{T}_j(s, a)))$$

Hence, the relationship between $\mathscr{H}_i(s, a)$ and $\mathscr{H}_j(s, a)$ depends on whether $\mathscr{H}_j(\mathscr{T}_j(s, a), \pi(\mathscr{T}_j(s, a)))$ equals $\mathscr{H}_i(\mathscr{T}_i(s, a), \pi_i(\mathscr{T}_i(s, a)))$. Due to insufficient information, we cannot definitively state their equality; thus, $\mathscr{H}_i(s, a) \lessgtr \mathscr{H}_j(s, a)$.

In the edge case scenario, if $\forall M_i, M_j \in \Omega_T$, $\mathscr{T}_i(s, a) \neq \mathscr{T}_j(s, a)$ but $\mathscr{H}_i(s, a) = \mathscr{H}_j(s, a)$, it implies that the distribution of tasks has negligible variance and is not worth investigating. However, if the distribution of dynamics is sufficiently large to create significant variability between tasks, then $\mathscr{H}_i(s, a) \lessgtr \mathscr{H}_j(s, a)$ holds, and therefore, each $M \in \Omega_T$ has its unique $\mathscr{H}$. □

Based on this lemma, we deduce that $\mathcal{H}(s, a)$ is a distribution such that:

$$\mathcal{H}(s, a) = \{\mathscr{H}_i(s, a) \mid \forall M_i \in \Omega_T\}$$

 doi:10.20944/preprints202502.0392.v1

Therefore, we can write:

$$\begin{aligned}\mathbb{E}_{M\sim\Omega_T}[Q^*(s,a)] &= \mathbb{E}_{M\sim\Omega_T}\mathbb{E}^*\left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t\gamma^t\right] \\ &= \mathbb{E}_{\mathscr{H}(s,a)\sim\mathcal{H}(s,a)}\mathbb{E}^*\left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t\gamma^t\right]\end{aligned}$$

This establishes that the expectation of the value function $Q^*(s,a)$ depends on the horizon distribution $\mathscr{H}$ sampled from the tasks distribution Ω_T. □

5.5. Value Function Distribution

We have established that the distribution of an MDP can be expressed in terms of the distribution of the path lengths generated by their policies. However, for transfer learning, our focus shifts to the distribution of the value function, as the agent will have access to the value function as part of the learning process, rather than the trajectory lengths. To formalize this, we define $\mathcal{Q}$ as the distribution of the value functions for all tasks within the distribution $\Omega \in \{\Omega_g, \Omega_T\}$. In this context, $\mathcal{Q} = \{Q_i | M_i \in \Omega\}$.

Consequently, the distribution of the value function can be represented as:

$$\mathcal{Q}(s,a) = \left\{ \mathbb{E}^*\left[\sum_{t=0}^{\mathscr{H}_i^*(s,a)} r_t\gamma^t\right] \;\middle|\; \mathscr{H}_i^*(s,a) \in \mathcal{H}(s,a) \right\}$$

While the general results apply to all MDPs, we now shift our attention to SBR-MDPs, which are characterized by binary reward structures.

Proposition 5. *Given MDP M_i where g is a terminal state, and reward function is a positive binary reward, using the bellman equation on the expected optimal path of the optimal policy π^*, the state-action value function is as follows:*

$$Q^*(s,a) = \gamma^{\mathscr{H}^*(s,a)} \prod_{t=0}^{\mathscr{H}^*(s,a)} T(s_t, a_t, s_{t+1}) \tag{1}$$

Proof. See Appendix A.2. □

Thus, the distribution of the value function can be written as:

$$\mathcal{Q}(s,a) = \left\{ \gamma^{\mathscr{H}_i^*(s,a)} \prod_{t=0}^{\mathscr{H}^*(s,a)} T(s_t, a_t, s_{t+1}) \mid \mathscr{H}_i^*(s,a) \in \mathcal{H}(s,a) \right\}.$$

6. Case Study: Log-Normality of the Value Function Distribution

To test the value function distribution and its connection to EOPL distributions, we test the case where EOPL follows a normal distribution. In RL, the normal distribution often emerges due to its presence in nature and its ability to formulate various random events and uncertainties within the task elements [5]. For example, many algorithms assume that rewards, transition dynamics or noise in the environment follow a Gaussian distribution, as it effectively models real-world variability [11]. This makes the normal distribution a natural fit for representing the EOPL.

Proposition 6. *If $\forall s, a \;\; \mathcal{H}(s,a) \equiv \mathcal{N}(\mu_{\mathcal{H}(s,a)}, \sigma^2_{\mathcal{H}(s,a)})$, then the value function distribution $\mathcal{Q}(s,a)$ follows a log-normal distribution such as:*

$$\ln \mathcal{Q}(s,a) \sim \mathcal{N}(\mu_Q(s,a), \sigma^2_Q(s,a))$$

 doi:10.20944/preprints202502.0392.v1

Proof. We apply the natural logarithm on both sides of equation (1)

$$\begin{aligned}\ln Q(s,a) &= \mathscr{H}^*(s,a)\ln\gamma + \ln(\prod_{t=0}^{\mathscr{H}^*(s,a)} T_t)\\ &= \mathscr{H}^*(s,a)\ln\gamma + \sum_{t=0}^{\mathscr{H}^*(s,a)} \ln T_t\end{aligned}$$

If we assume $\mathcal{H}^*(s,a) \equiv \mathcal{N}(\mu,\sigma^2)$, then we can assume under certain conditions that $\sum_{t=0}^{\mathscr{H}^*(s,a)} \ln T_t$ follows normal distribution using central limit theorem. For full proof check Appendix A.3. Finally we write:

$$\ln \mathcal{Q}(s,a) \sim \mathcal{N}(\mu_Q(s,a), \sigma_Q^2(s,a))$$

where the parameters $\mu_Q(s,a)$ and $\sigma_Q^2(s,a)$ depend on the scaling Consequently, $\mathcal{Q}(s,a)$ follows a log-normal distribution. □

Algorithm: LogQInit

The log-normal distribution properties suggests two natural expectation strategies mean and median. Accordingly, we introduce two approaches loqinit as follows:

1. LogQInit-Median: The median, being robust to skewness, is calculated as:

$$Q^{\oslash}(s,a) := e^{\mu(s,a)}$$

2. LogQInit-Mean: The mean, which reflects the average expected value, is given by:

$$Q^{\oslash}(s,a) := e^{\mu(s,a)+\frac{1}{2}\sigma^2(s,a)}$$

Algorithm 1 outlines the LogQInit method, which proceeds as follows:

- The agent first samples a subset of tasks which are learnt sufficiently to get a close approximation of their optimal function Q^* and are stored in $\mathcal{Q}$
- the EOPL distribution parameters for each state action $(\mu_Q(s,a), \sigma_Q^2(s,a))$ is then estimated using the sample
- Using these estimates, the value function is initialized and learnt afterwards for subsequent tasks and

 doi:10.20944/preprints202502.0392.v1

Algorithm 1: LOGQINIT

variables: $\Omega; \sigma : S \times A$; $\mu : S \times A$;
n : number of training tasks
for $i = 1$ **to** n **do**
  sample new task $M_i \sim \Omega$
  learn$Q^*(M_i)$
  $\mathcal{Q} \cup Q_i^*$
end for
`***Estimate σ and μ***`
for all state-actions s, a **do**
  $\sigma^2(s,a) \leftarrow Variance(\ln \mathcal{Q}(s,a))$
  $\mu(s,a) \leftarrow Mean(\ln \mathcal{Q}(s,a))$
end for
`***Starting lifelong learning***`
repeat
  sample new task $M_{new} \sim \Omega$
  initialization : $Q_{new}^{\emptyset} \leftarrow e^{\mu(s,a)}$(median) or $e^{\mu(s,a)+\frac{\sigma^2(s,a)}{2}}$ (mean)
  learn
until Agent lifetime

7. Experiments

The objectives of our experiments are summarized in two key points: Firstly, we aim to demonstrate that the distribution of tasks can be translated into distinct distributions of optimal paths for every state-action value, in both cases of goal distributions and dynamics distributions. Secondly, we seek to test our hypothesis that, in the case of a normal task distribution, using the log-normal parameters for initialization (LOGQINIT) is more effective than existing initialization methods. To evaluate this, we compare LOGQINIT with the following strategies:

- MAXQINIT [1]: A method based on the optimistic initialization concept, where each state-action value is initialized using the maximum value from previously solved tasks.
- UCOI [19]: This method balances optimism in the face of uncertainty by initializing with the maximum value, while relying on the mean when certainty is higher.

We also compare with other popular transfer methods which are:

- PTRS (Policy Transfer with Reward Shaping) [7]: This approach utilizes the value function from the source task as a potential function to shape the reward in the target task, expressed as $r'(s,a) = r(s,a) + \gamma\phi(s') - \phi(s)$. Here, the potential function is defined as the mean of the set $\mathcal{Q}$.
- OPS-TL [15]: This method maintains a dictionary of previously learned policies and employs a probabilistic selection mechanism to reuse these policies for new tasks.
- DRS (Distance-Based Reward Shaping): We introduce this algorithm as a straightforward reward-shaping method that relies on the distance between the current state and the goal state.

Additionally, we compare these approaches with Q^*, which we claim we sufficiently obtained through Q-learning, and a non-transfer baseline, where the Q-function was initialized by zeros.

7.1. Goal Distribution: Gridworld

The first experiment investigates the normal distribution of goals. In the standard Gridworld environment which is commonly employed in RL theories [25]. The Gridworld is a 2D map of states identified by its coordinates $[i, j]$ which enables straightforward determination of EOPL typically quantified by the Manhattan distance from any state to the goal. such a setting offers simplicity and efficacy in demonstrating theoretical propositions and visualizing the different experiment results.

 doi:10.20944/preprints202502.0392.v1

7.1.1. Environment Description:

The environment is a 30×30 Gridworld where each state is identified by its coordinates $[i, j]$. The agent can execute four actions (up, down, left, and right) across all states. When the agent takes an action, it transitions to the next state with a probability of $p = \max_{s'} T(s, a, s') \sim uniform(0.6, 0.9)$ while other states have the probability of $\frac{1-p}{3}$. A reward of 1 is assigned to the terminal state, denoted as g, while all other states in the environment yield a reward of 0. Tasks are generated with diverse goals defined as $g = [i \sim \mathcal{N}(20, \sigma^2), j \sim \mathcal{N}(20, \sigma^2)]$. Various distributions are tested based on different variance values, specifically $\sigma^2 \in [1, 3, 5, 7, 10]$ besides a uniform distribution where $g[i \sim uniform(0, 30), j \sim uniform(0, 30)]$. The goal state's normal distribution leads with a normal distribution of distances from each state to the goal , and therefore the normality of EOPLs distribution.

7.1.2. Results and Discussion:

Log-normality of value function: We initially trained the agent on 100 tasks drawn from the given distribution. These tasks were learned from scratch and assumed to be learned until completion. Figure 5 illustrates the histogram of a set of $Q(s, a)$ values of the states $s = \{[0, 0], [20, 20], [29, 29]\}$ obtained from learning these 100 tasks drawn from the distribution with the variance $\sigma^2 = 7$. By examining the histograms of $s = [0, 0]$ and $s = [29, 29]$ (respectively Figures 5(d) and 5(e)), we observe that the normal distribution of goals indeed resulted in a log-normal distribution of $Q(s, a)$ with negligible divergence, likely stemming from the function approximation. Except for $s = [20, 20]$, where the distribution of $Q(s, a)$ is itself normal since the goal is centred on this state. However, we argue that the choice of initialization values will not significantly impact learning. This is because the assigned value is high enough to prompt the agent to explore each state in this area, and the goal is likely to appear there, ensuring that exploration is not wasted.

 doi:10.20944/preprints202502.0392.v1

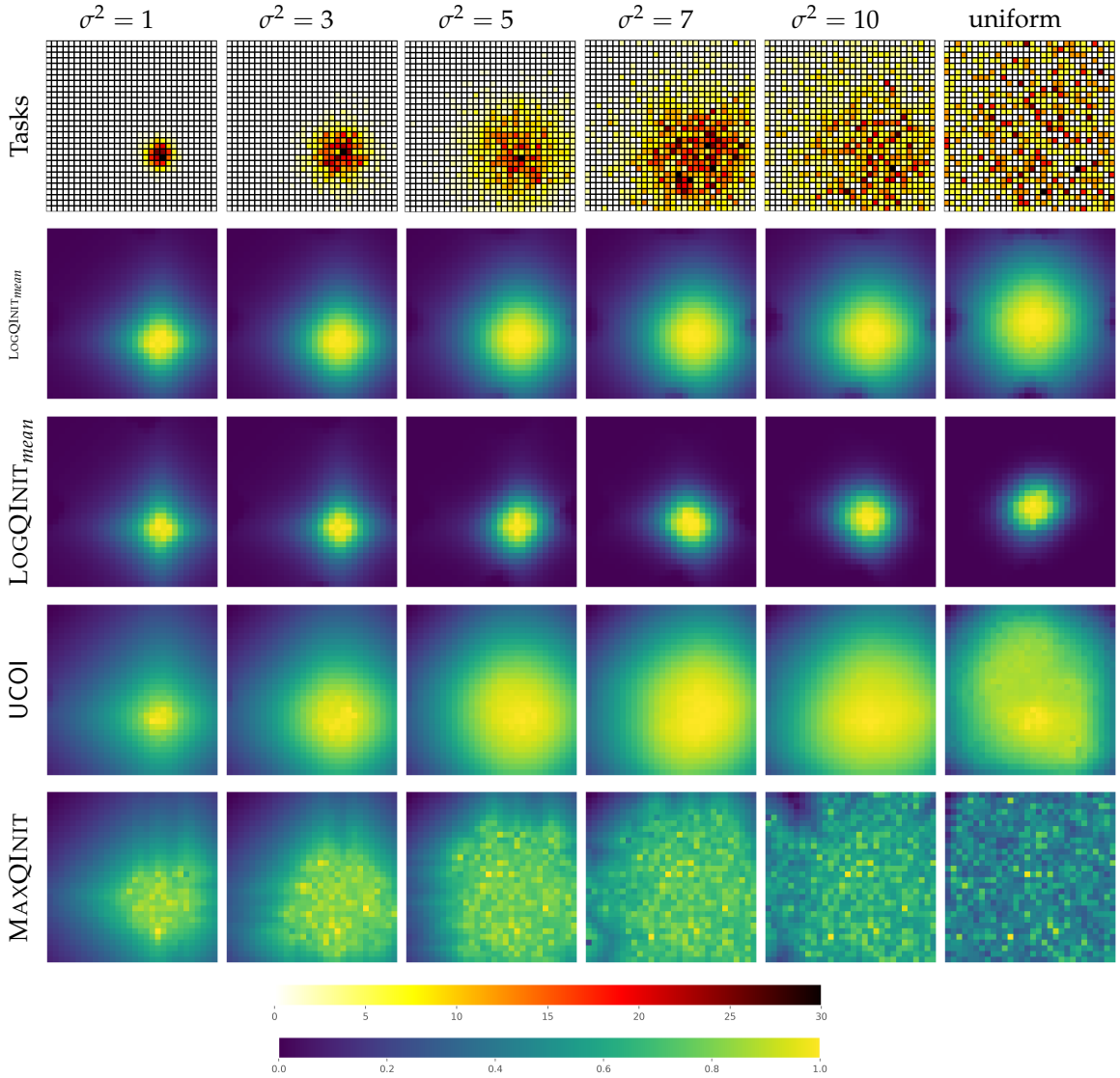
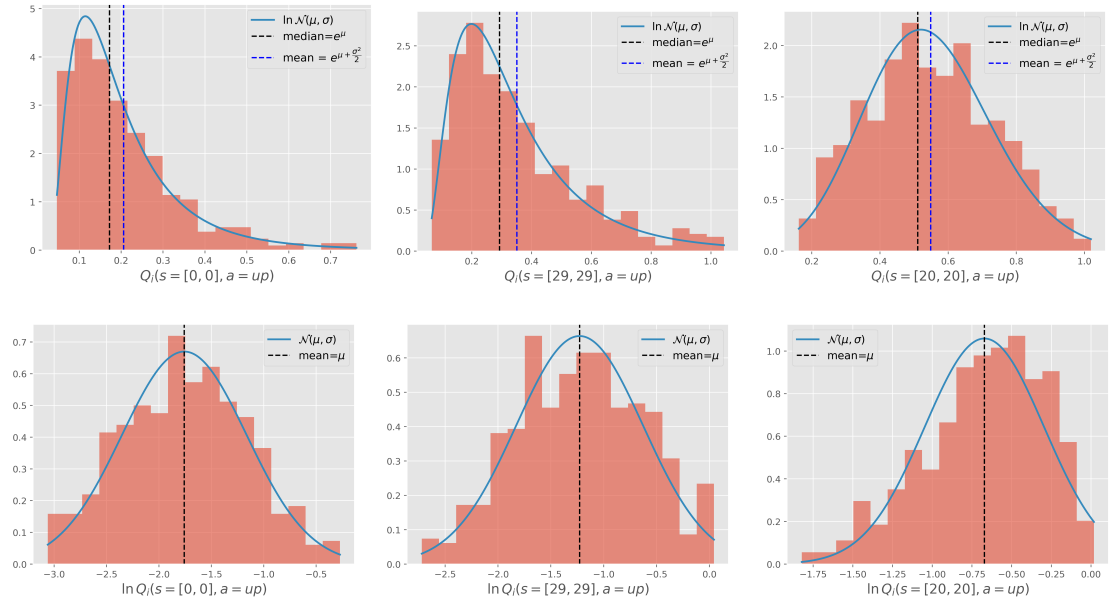


Figure 4. Heatmaps of the initial value functions in the GridWorld environment under various distribution scenarios.

LogQInit vs initialization methods: The second phase of our analysis involves a comprehensive comparison between our proposed approach and existing methodologies. To begin, we visualize the initialized function and heatmaps, as depicted in Figure 4. Each column in the figure corresponds to a specific distribution with different variance, with the last column representing the uniform distribution. The first row represents the heatmap of goal distribution in the gridworld environment, and the others rows each corresponds to an initialization method.

 doi:10.20944/preprints202502.0392.v1



(d) $s = [0,0]$ (e) $s = [29,29]$ (f) $s = [20,20]$ (edge case)

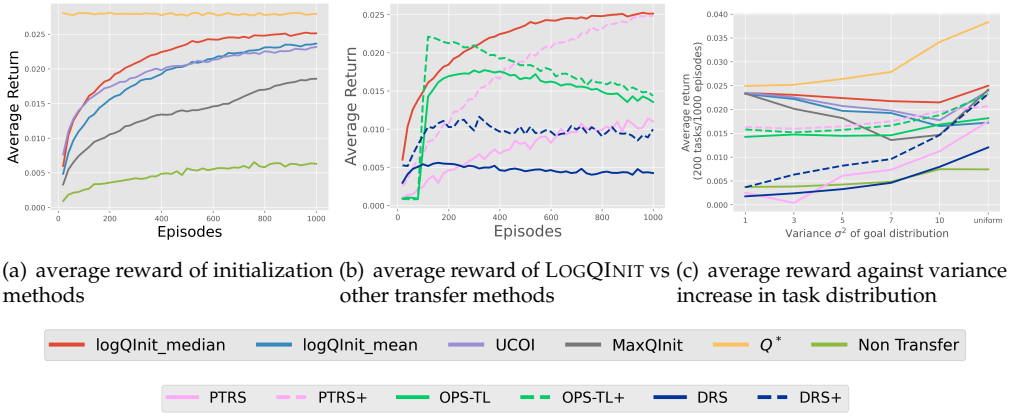
Figure 5. (Top Row) Histogram of $\mathcal{Q}(s,a)$ for 100 GridWorld tasks learned to completion; (Bottom Row) Corresponding histogram of $\ln Q(s,a)$.

Our findings reveal that our approach, LogQInit, generates a heatmap closely aligned with the task distribution. In essence, high values are centred around the centre of the goal distribution, and the area of high values gradually expands with the increase in the variance of the distribution. On the contrary, UCOI, though similar to LogQInit in assigning high values to areas with frequent goal appearances, exhibits a wider distribution of high values in scenarios with high variance. This implies that the agent may end up exploring more than it does with LogQInit. On the other hand, as previously demonstrated by [19], MaxQInit showcases an irregular pattern in the initial value function. While the heatmap resembles other approaches for $\sigma^2 = 1$, it becomes increasingly random and chaotic as σ^2 increases. This is attributed to two factors: MaxQInit follows an optimistic initialization that assigns the maximum value, and the sample size used is only 100, whereas the condition for MaxQInit is to use pmin, resulting in approximately 4754 computations, which is impractical and computationally intensive.

The deduced results from the heatmaps are backed up by the comparison of average rewards obtained from learning 200 tasks using these approaches, as depicted in Figure 6. It is evident that LogQInit outperforms other approaches for all distributions, except when $\sigma^2 = 1$ since the initial value functions are basically similar. LogQInit-mean ranks third due to its lack of allowance for exploration, and MaxQInit yields the least favourable results among the compared approaches.

LogQInit vs transfer methods: Figure 6(b) shows a comparison between LogQInit-median and other ptrs opstl and drs, where + besides the latter methods indicate that the value function was initialized by loqinit-median , the absence of + indicates it was initialized by zero. notably LogQInit shows better performance

 doi:10.20944/preprints202502.0392.v1



(a) average reward of initialization methods (b) average reward of LOGQINIT vs other transfer methods (c) average reward against variance increase in task distribution

Figure 6. Average Reward from Transfer Methods in Gridworld: Left - Average reward of initialization methods for the distribution of goals with $\sigma^2 = 7$; Middle - LOGQINIT median compared with other transfer methods for the distribution of goals with $\sigma^2 = 7$; Right - Average reward over 1000 episodes across different distributions.

7.2. Dynamics Distribution: Mountain Car

The second experiments are directed for dynamics distribution in the environment, we chose the Mountain Car, a classic control problem introduced in the field of reinforcement learning.

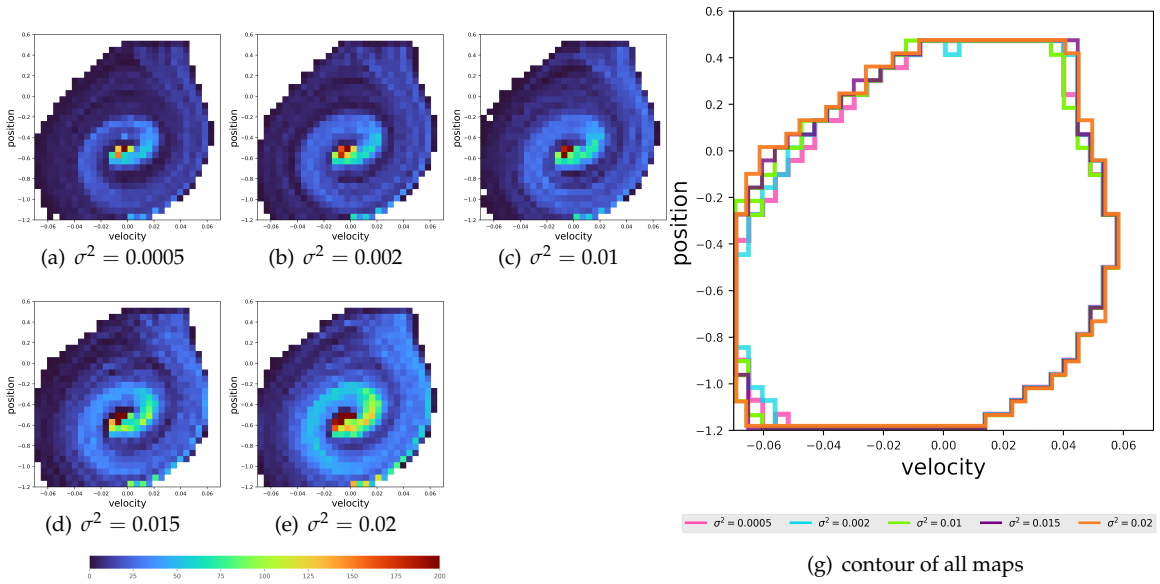
7.2.1. Environment Description:

The car is situated in a valley between two hills, with the objective of reaching a flag positioned at the peak of one of the hills. The state space of the Mountain Car environment is defined by the car's position $\in [-1.2, 0.6]$ and velocity $\in [-0.07, 0.07]$. The action space is represented as $A = \{0, 1, 2\}$, where 0 corresponds to doing nothing, 1 represents accelerating forward, and 2 represents decelerating or accelerating backwards. To illustrate a distribution of dynamics, we add a normal distribution of noise to the environment, particularly in the right hill area, as depicted in the Figure A1. The dynamics are modeled as $position_{t+1} = position_t + velocity_t + \eta$, where $\eta \sim \mathcal{N}(0, \sigma^2)$. The variance σ^2, takes different values in various distribution scenarios, such as $\sigma^2 \in [0.0002, 0.005, 0.01, 0.015, 0.02]$.

7.2.2. Results and Discussion:

Normality of EOPL distribution: We initiated the experiment by learning 100 tasks independently for each distribution without transfer for $5e^5$ episodes using a tabular Q-learning (the detail are in the Appendix A.4). Firstly, we want to check the EOPLs resulting from the 100 tasks and whether it abides by the propostion 4. Therefore, we test the final policy starting from state s_0 tasks, we counted the occurrences of all the possible states within the path generated by the policy. These counts are in the shape of a grid the tuple $[position, velocity]$ which is visualized as a heatmap in Figure 7. Notably, certain states manifest more frequently in distributions with larger variances compared to those with smaller variances . This observation indicates that the changing of this kind of dynamics leads to the different frequency of some states due to the difference of their function $\mathscr{T}$ and so are sequences of states in the EOPL across tasks. Next, to prove that EOPLs differs also in lengths, displayed the EOPLs from the 100 tasks as histograms accompanied by its corresponding density function in Figure 8. Remarkably, we observe that the distribution of the number of steps closely adheres to a normal distribution, aligning with our initial hypothesis. To compare the various distributions, we present their density functions in a unified plot, as illustrated in Figure 8. Notably, all distributions exhibit nearly identical means, corresponding to the number of steps obtained in an environment without noise. Moreover, we observe that the larger the variance of the distribution, the wider the variance of EOPL distribution.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 February 2025 doi:10.20944/preprints202502.0392.v1

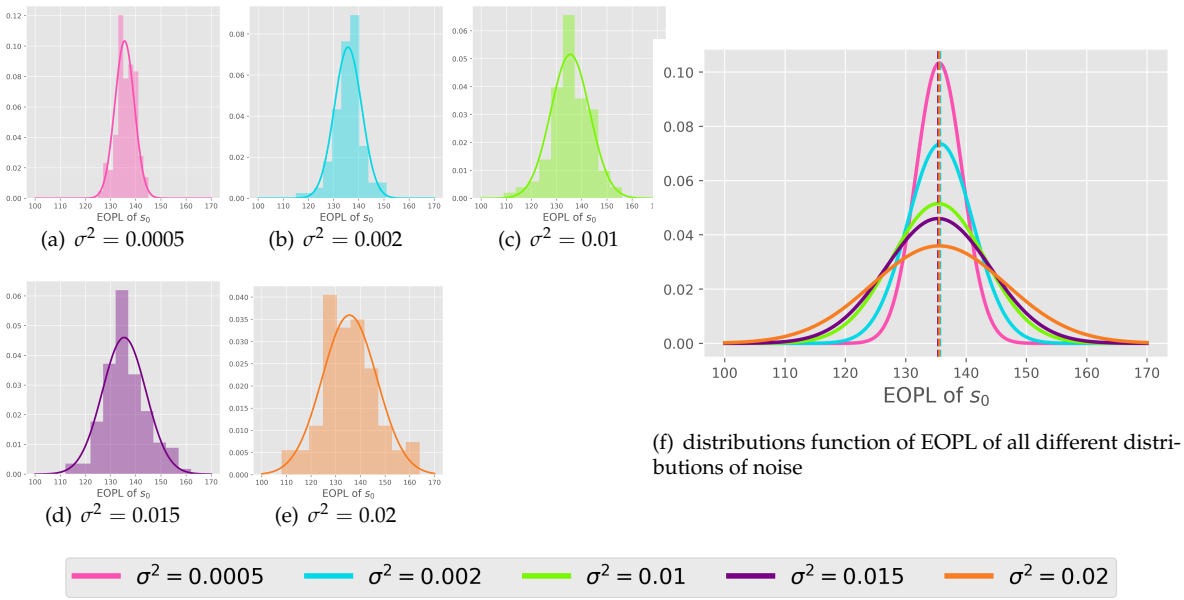


(a) $\sigma^2 = 0.0005$ (b) $\sigma^2 = 0.002$ (c) $\sigma^2 = 0.01$

(d) $\sigma^2 = 0.015$ (e) $\sigma^2 = 0.02$

(g) contour of all maps

Figure 7. Visualization of all the states occurrences in the EOPL across 100 tasks. The x-axis represents velocity, and the y-axis represents position. Left: heatmaps showing the occurrence frequency for each distribution. Right: a close-up view of the heatmaps' outlines of all the distributions.



(a) $\sigma^2 = 0.0005$ (b) $\sigma^2 = 0.002$ (c) $\sigma^2 = 0.01$

(d) $\sigma^2 = 0.015$ (e) $\sigma^2 = 0.02$

(f) distributions function of EOPL of all different distributions of noise

Figure 8. Histogram of EOPLs of the state $s_0 = [p = -0.5, v = 0]$ across the 100 tasks. left: the histogram of each distribution. right: the plot of all density functions of all the distribution of the noise together.

log normality of value function: We have displayed the histogram of the value function in Figure 10, for the states [position = -0.5,velocity = 0.0] corresponding to the starting state, [position= -0.3 ,velocity =0.02] corresponding to a state in the middle of the path, and [position =0.2 ,velocity=0.04] corresponding to a state that reaches the target. We notice that the histogram of 100 tasks of these functions follows a log distribution and normal distribution just as hypothesized, except for the third state which corresponds to the edge case explained in Appendix A.3.

LOGQINIT Performance: Finally, Figure 9 presents the average return obtained using different initialization methods. The results indicate that LOGQINIT-median, followed by UCOI, achieved the best performance in environments with lower variance in the noise distribution. In contrast, LOGQINIT-mean performed better in higher-variance settings, likely due to the need for more cautious exploration

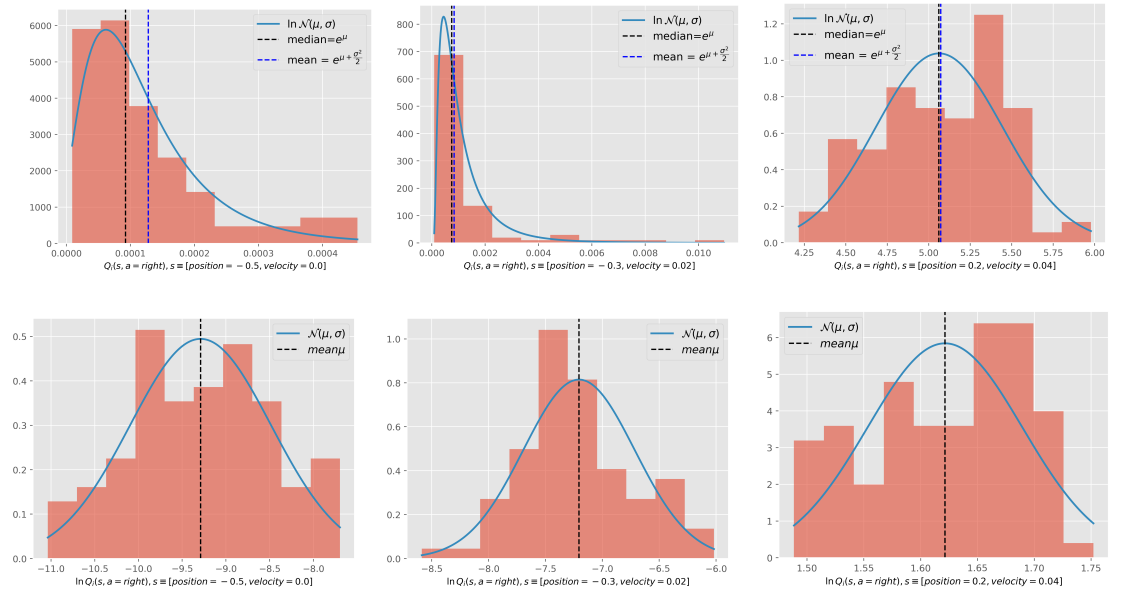
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 February 2025 doi:10.20944/preprints202502.0392.v1

in such environments. However, we did not include a comparison with other transfer methods, as they provided no measurable improvement over 1000 episodes.

(a) average reward of distribution $sigma^2 = 0.01$

(b) average reward of each distribution of noise

Figure 9. Average reward of 100 tasks of Mountain Car environment using different transfer methods.



(d) state =$[p = -0.5, v = 0]$ (e) state =$[p = -0.3, v = 0.02]$ (f) state =$[p = 0.2, v = 0.04]$(edge case)

Figure 10. Histogram of $Q(s, a)$ of 100 tasks ran into completion (top row) and its correspondent set of ln $Q(s, a)$ (lower row) for the states. $[p, v]$ is abbreviation of $[position, velocity]$.

8. Conclusion

In this paper, we examined the distribution of state-action value functions induced by a distribution of tasks. Specifically, we studied tasks with the same state-action space but differing in dynamics or rewards. We demonstrated that the value functions of these MDPs can be directly expressed in terms of state-action-specific finite horizons by extracting the Expected Optimal Path Length (EOPL) of each state-action pair. This formulation established a direct connection between the task distribution and the resulting distribution of the value function.

We focused MDPs with terminal goal and sparse positive rewards, as this setting is widely used in the literature. This MDP configuration revealed an exponential relationship between the value function and EOPLs.

To validate our propositions, we examined the case where the task distribution follows a normal distribution. We showed that this results in a log-normal distribution of value functions, leading to the proposal of **LogQInit**, an initialization method based on the median and mean of the value function distribution. We tested this approach on task distributions where the resulting EOPLs are normally

 doi:10.20944/preprints202502.0392.v1

distributed. The results confirmed our hypothesis and showed that **LogQInit** provides a more accurate initialization than existing methods.

However, our work has two main limitations. First, we only tested normal and uniform distributions. In real-world applications, the distribution of optimal path lengths may be more complex, especially in environments where the dynamics function does not transition states linearly. Extending this study to other types of distributions could further improve our approach. Second, our initialization method was evaluated in a tabular setting. A potential improvement is to discretize the state-action space and adapt the initialization process to continuous state spaces.

Appendix A

Appendix A.1. Proof of Proposition 1

Proof. To prove that in an MDP with a terminal goal and positive rewards, the value function can be expressed as:

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \mathbb{E}\left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t\gamma^t\right]$$

we need to show that $\mathscr{H}^*(s,a) = \min_\pi \mathscr{H}^\pi(s,a)$, which implies:

$$\forall\pi,\quad \mathbb{E}\left[\sum_{t=0}^{\mathscr{H}^*(s,a)} r_t\gamma^t\right] \geq \mathbb{E}\left[\sum_{t=0}^{\mathscr{H}^\pi(s,a)} r_t\gamma^t\right].$$

Assume there exists a policy π^+ such that $\mathscr{H}^{\pi^+}(s,a) < \mathscr{H}^{\pi^*}(s,a)$.

Since the goal state g is terminal, no rewards are collected beyond reaching g. Therefore, for $t > \mathscr{H}(s,a)$, we have $r(s_t,a_t) = 0$. This allows us to rewrite the infinite-horizon expectation as:

$$\mathbb{E}^*\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \mathbb{E}^*\left[\sum_{t=0}^{\mathscr{H}^*(s,a)}\gamma^t r(s_t,a_t) + \sum_{t=\mathscr{H}^*(s,a)+1}^{\infty}\gamma^t\cdot 0\right]$$

and

$$\mathbb{E}^+\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \mathbb{E}^+\left[\sum_{t=0}^{\mathscr{H}^{\pi^+}(s,a)}\gamma^t r(s_t,a_t) + \sum_{t=\mathscr{H}^{\pi^+}(s,a)+1}^{\infty}\gamma^t\cdot 0\right].$$

Since $\mathscr{H}^{\pi^+}(s,a) < \mathscr{H}^{\pi^*}(s,a)$, it follows that:

$$\mathbb{E}^{\pi^+}\left[\sum_{t=0}^{\mathscr{H}^{\pi^+}(s,a)} r_t\gamma^t\right] > \mathbb{E}^{\pi^*}\left[\sum_{t=0}^{\mathscr{H}^{\pi^*}(s,a)} r_t\gamma^t\right].$$

This contradicts the definition of Q^*, which states that the optimal policy maximizes the expected return. Hence, we conclude that:

$$\mathscr{H}^*(s,a) = \min_\pi \mathscr{H}^\pi(s,a).$$

Appendix A.2. Proof of Proposition 5

To prove propostion 5, we use the bellman optimality equation since we are dealing with the optimal function Q^* which is consistent :

$$Q*(s,a) = r(s,a,s') + \gamma T(s,a,s')\max_{a'} Q^*(s',a')$$

 doi:10.20944/preprints202502.0392.v1

We set that s, a on the left side of the equation to be starting from time $t = 0$. the EOPL of the state s_0 and action a_0, with $s_{t+1} = \mathcal{T}(s_t, \pi(s_t))$ is :

$$\tau^*(\pi^*) = s_0, a_0, s_1, \pi^*(s_1), ..., s_{\mathscr{H}^*(s_0,a_0)-1}, \pi^*(s_{\mathscr{H}^*(s_0,a_0)-1}), g$$

By definition we have $\pi^*(s) = argmax_a Q^*(s,a)$ therefore $max_a Q^*(s,a) = Q^*(s, \pi^*(s))$ we run the bellman equation on the sequence $\tau^*(\pi^*)$, specifying that $r_t = r(s_t, a_t, s_{t+1})$ and $T_t = T(s_t, a_t, s_{t+1})$

$$\begin{aligned}
Q^*(s_0, a_0) &= r_0 + \gamma T_0 Q^*(s_1, \pi^*(s_1)) \\
&= r_0 + \gamma T_0 \Big(r_1 + \gamma T_1 \Big(r_2 + \gamma T_2 \Big(... \Big(r_{\mathscr{H}^*(s_0,a_0)} + \gamma T_{\mathscr{H}^*(s_0,a_0)-1} r_g \Big) ... \Big) \Big) \Big) \\
&= \gamma T_0 \Big(\gamma T_1 \Big(\gamma T_2 \Big(... \Big(\gamma T_{\mathscr{H}^*(s_0,a_0)} \Big) ... \Big) \Big) \Big) \text{by substracting using positive sparse reward} \\
&= \gamma^{\mathscr{H}^*(s_0,a_0)} \prod_{t=0}^{\mathscr{H}^*(s_0,a_0)} T_t
\end{aligned}$$

□

Appendix A.3. Proof of normality of $\sum_{t=0}^{\mathscr{H}(s,a)} \ln T_t$

Proof. We consider the transitions $T_1, T_2, \ldots, T_{\mathscr{H}(s,a)}$ to be a sequence of i.i.d. random variables such. We have $\mathscr{H}(s,a)$ as a random variable representing the number of terms in the product, independent of T_t, and assume $\mathscr{H}(s,a) \in \mathbb{N}$. Therefore, the sum of the logarithms $\sum_{t=0}^{\mathscr{H}^*(s,a)} \ln T_t$ approximately follows a normal distribution under the following conditions:

1. Finite Expectation and Variance of $\ln T_t$: The expectation $\mathbb{E}[\ln T_t]$ depends on the distribution of T_t, with a higher probability density near 1 leading to a less negative mean. The variance $\mathrm{Var}(\ln T_t)$ is finite and captures the spread of $\ln T_t$:
   we know that for sure that $T_t \in (a, 1]$, with $0 \ll a \leq 1$ since this transition is $\max_{s,a} T(s,a,s')$. Such behavior is characteristic of deterministic or near-deterministic environments, which are common in many RL tasks. Based on this observation, we posit that RL environments of interest adhere to this description, thereby satisfying the condition of finite expectation and variance for $\ln T_t$.
2. Sufficiently Large $\mathbb{E}[\mathscr{H}^*(s,a)]$: When Y is large enough (e.g., $\mathbb{E}[\mathscr{H}^*(s,a)] \gg 1$), the sum $\sum_{t=0}^{\mathscr{H}(s,a)} \ln T_t$ satisfies the Central Limit Theorem, approximating a normal distribution due to the aggregation of independent terms.

Edge Case: if $\mathscr{H}^*(s,a)$ is small, the normality of $\sum_{t=1}^{\mathscr{H}^*(s,a)} \ln T_t$ may break down. This happens when the state-action pair in question is often close to the goal, leading to fewer steps to the goal. In such cases, breaking the normality is not problematic, because the values of value function around these states are overall high and so is the expectation from this set of values, which is easy for the agent to converge during learning according to optimistic initialization concept. Consequently, this edge case does not interfere with the objective of value function initialization. □

Appendix A.4. Environment Description

Mountain car: to learn it in a tabular fashion we discretized the state position and velocity interval into 30x30. we used a binary reward where zero is assigned to non rewarding states and 1 to achieving the goal , in order to fit into the logarithm properties used by LogQInit. the learning rate alpha is $\alpha = 0.05$ and the exploration method we used is ϵ-greedy where $\epsilon = 0.5$ during learning tasks without transfer, and $\epsilon = 0.1$ for tasks with receiving knowledge transfer.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 February 2025 doi:10.20944/preprints202502.0392.v1

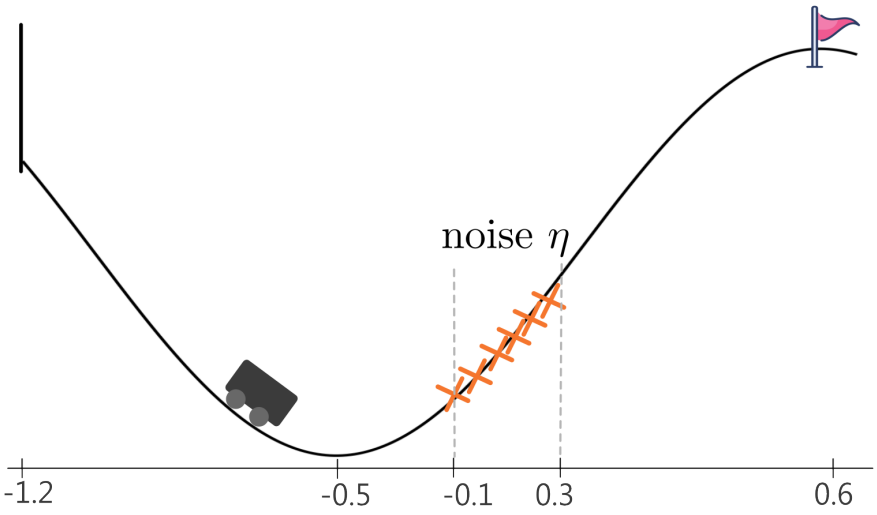
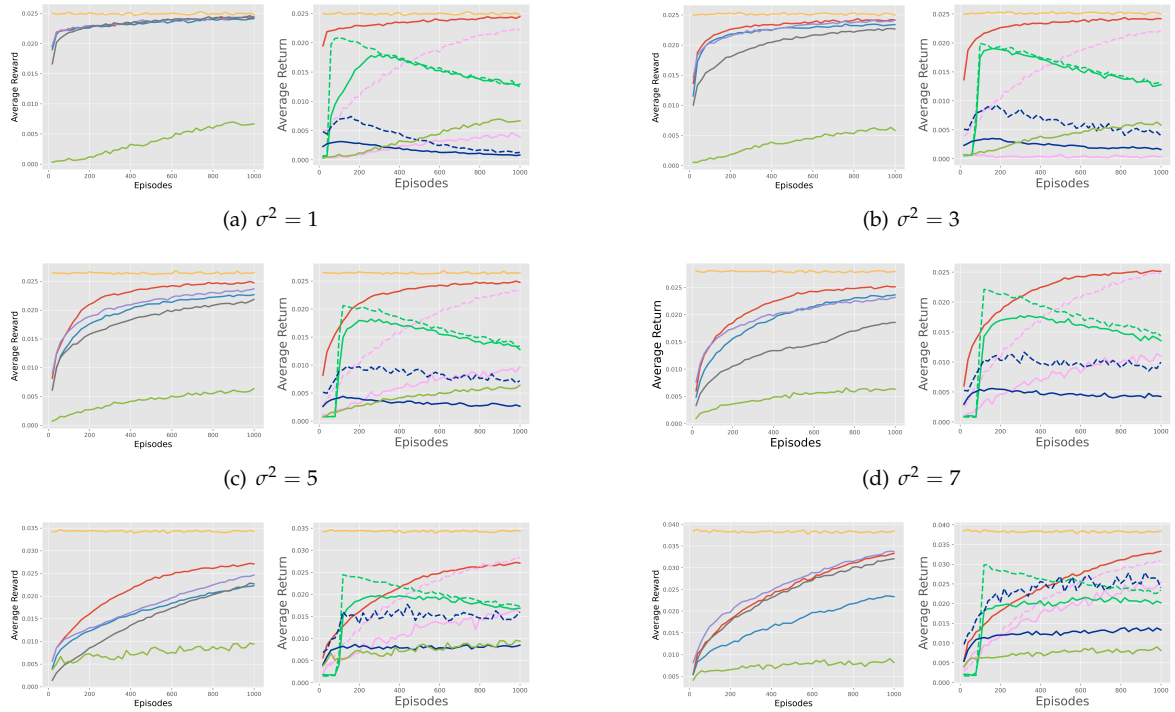


Figure A1. Mountain Car environment with added noise on the hill.



(a) $\sigma^2 = 1$ (b) $\sigma^2 = 3$

(c) $\sigma^2 = 5$ (d) $\sigma^2 = 7$

(e) $\sigma^2 = 10$ (f) $uniform$

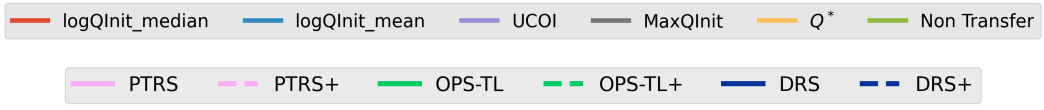
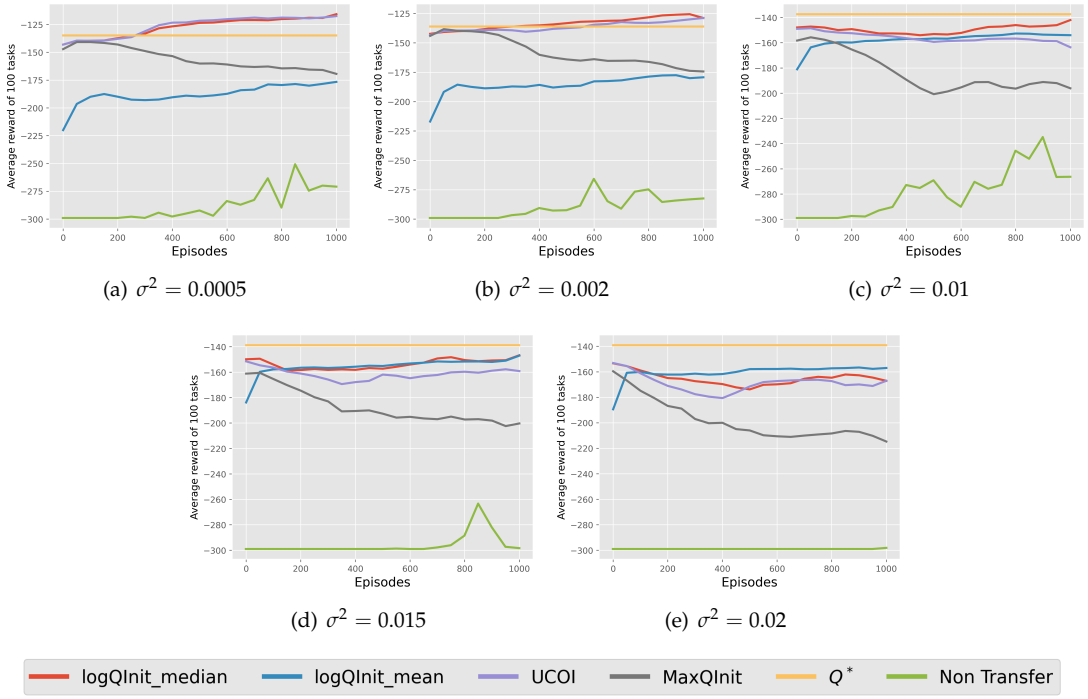


Figure A2. average return of on Gridworld using the different initialization methods and transfer methods.

 doi:10.20944/preprints202502.0392.v1



(a) $\sigma^2 = 0.0005$ (b) $\sigma^2 = 0.002$ (c) $\sigma^2 = 0.01$

(d) $\sigma^2 = 0.015$ (e) $\sigma^2 = 0.02$

Figure A3. average return of the different initialization methods performance in mountain car environment across different noise distribution.

References

1. Abel, D., Jinnai, Y., Guo, S., Konidaris, G. & Littman, M. Policy and value transfer in lifelong reinforcement learning. *International Conference On Machine Learning*. pp. 20-29 (2018)
2. Agrawal, P. & Agrawal, S. Optimistic Q-learning for average reward and episodic reinforcement learning. *ArXiv Preprint ArXiv:2407.13743*. (2024)
3. Ajay, A., Gupta, A., Ghosh, D., Levine, S. & Agrawal, P. Distributionally Adaptive Meta Reinforcement Learning. *Advances In Neural Information Processing Systems*. **35** pp. 25856-25869 (2022)
4. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O. & Zaremba, W. Hindsight experience replay. *Advances In Neural Information Processing Systems*. **30** (2017)
5. Bellemare, M., Dabney, W. & Munos, R. A distributional perspective on reinforcement learning. *International Conference On Machine Learning*. pp. 449-458 (2017)
6. Bi, Z., Guo, X., Wang, J., Qin, S. & Liu, G. Deep Reinforcement Learning for Truck-Drone Delivery Problem. *Drones*. **7** (2023), https://www.mdpi.com/2504-446X/7/7/445
7. Brys, T., Harutyunyan, A., Taylor, M. & Nowé, A. Policy Transfer using Reward Shaping.. *AAMAS*. pp. 181-188 (2015)
8. Castro, P. & Precup, D. Using bisimulation for policy transfer in MDPs. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **24** pp. 1065-1070 (2010)
9. D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M. & Peters, J. Sharing knowledge in multi-task deep reinforcement learning. *ArXiv Preprint ArXiv:2401.09561*. (2024)
10. Guo, Y., Gao, J., Wu, Z., Shi, C. & Chen, J. Reinforcement learning with Demonstrations from Mismatched Task under Sparse Reward. *Conference On Robot Learning*. pp. 1146-1156 (2023)
11. Johannink, T., Bahl, S., Nair, A., Luo, J., Kumar, A., Loskyll, M., Ojea, J., Solowjow, E. & Levine, S. Residual reinforcement learning for robot control. *2019 International Conference On Robotics And Automation (ICRA)*. pp. 6023-6029 (2019)
12. Khetarpal, K., Riemer, M., Rish, I. & Precup, D. Towards continual reinforcement learning: A review and perspectives. *Journal Of Artificial Intelligence Research*. **75** pp. 1401-1476 (2022)
13. Lecarpentier, E., Abel, D., Asadi, K., Jinnai, Y., Rachelson, E. & Littman, M. Lipschitz lifelong reinforcement learning. *ArXiv Preprint ArXiv:2001.05411*. (2020)
14. Levine, A. & Feizi, S. Goal-conditioned Q-learning as knowledge distillation. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **37**, 8500-8509 (2023)

15. Li, S. & Zhang, C. An optimal online method of selecting source policies for reinforcement learning. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **32(1)** (2018)
16. Liu, M., Zhu, M. & Zhang, W. Goal-conditioned reinforcement learning: Problems and solutions. *ArXiv Preprint ArXiv:2201.08299*. (2022)
17. Liu, R., Shin, H. & Tsourdos, A. Edge-enhanced attentions for drone delivery in presence of winds and recharging stations. *Journal Of Aerospace Information Systems*. **20**, 216-228 (2023)
18. Lobel, S., Gottesman, O., Allen, C., Bagaria, A. & Konidaris, G. Optimistic initialization for exploration in continuous control. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **36(7)** pp. 7612-7619 (2022)
19. Mehimeh, S., Tang, X. & Zhao, W. Value function optimistic initialization with uncertainty and confidence awareness in lifelong reinforcement learning. *Knowledge-Based Systems*. **280** pp. 111036 (2023)
20. Mezghani, L., Sukhbaatar, S., Bojanowski, P., Lazaric, A. & Alahari, K. Learning goal-conditioned policies offline with self-supervised reward shaping. *Conference On Robot Learning*. pp. 1401-1410 (2023)
21. Padakandla, S. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*. **54**, 1-25 (2021)
22. Rakelly, K., Zhou, A., Finn, C., Levine, S. & Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *International Conference On Machine Learning*. pp. 5331-5340 (2019)
23. Salvato, E., Fenu, G., Medvet, E. & Pellegrino, F. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*. **9** pp. 153171-153187 (2021)
24. Strehl, A., Li, L. & Littman, M. Reinforcement Learning in Finite MDPs: PAC Analysis.. *Journal Of Machine Learning Research*. **10** (2009)
25. Sutton, R. & Barto, A. Reinforcement learning: An introduction. (MIT press,2018)
26. Taylor, M. & Stone, P. Transfer learning for reinforcement learning domains: A survey.. *Journal Of Machine Learning Research*. **10** (2009)
27. Tirinzoni, A., Sessa, A., Pirotta, M. & Restelli, M. Importance weighted transfer of samples in reinforcement learning. *International Conference On Machine Learning*. pp. 4936-4945 (2018)
28. Uchendu, I., Xiao, T., Lu, Y., Zhu, B., Yan, M., Simon, J., Bennice, M., Fu, C., Ma, C., Jiao, J. & Others Jump-start reinforcement learning. *ArXiv Preprint ArXiv:2204.02372*. (2022)
29. Vuong, T., Nguyen, D., Nguyen, T., Bui, C., Kieu, H., Ta, V., Tran, Q. & Le, T. Sharing experience in multitask reinforcement learning. *Proceedings Of The 28th International Joint Conference On Artificial Intelligence*. pp. 3642-3648 (2019)
30. Wang, J., Zhang, J., Jiang, H., Zhang, J., Wang, L. & Zhang, C. Offline meta reinforcement learning with in-distribution online adaptation. *International Conference On Machine Learning*. pp. 36626-36669 (2023)
31. Zang, H., Li, X., Zhang, L., Liu, Y., Sun, B., Islam, R., Combes, R. & Laroche, R. Understanding and addressing the pitfalls of bisimulation-based representations in offline reinforcement learning. *Advances In Neural Information Processing Systems*. **36** (2024)
32. Zhai, Y., Baek, C., Zhou, Z., Jiao, J. & Ma, Y. Computational benefits of intermediate rewards for goal-reaching policy learning. *Journal Of Artificial Intelligence Research*. **73** pp. 847-896 (2022)
33. Zhu, T., Qiu, Y., Zhou, H. & Li, J. Towards Long-delayed Sparsity: Learning a Better Transformer through Reward Redistribution.. *IJCAI*. pp. 4693-4701 (2023)
34. Zhu, Z., Lin, K., Jain, A. & Zhou, J. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. (2023)
35. Zou, H., Ren, T., Yan, D., Su, H. & Zhu, J. Learning task-distribution reward shaping with meta-learning. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **35(12)** pp. 11210-11218 (2021)